FEBRERO 1989
REVISTA PARA USUARIOS DE LA NORMA

LOAD MSX

AÑO 3- Nº 34 ₳ 45 - REP. ARGENTINA

MUSICA POR COMPUTADORA

INEDITOS

3 UTILITARIOS

SORTEOS y CONCURSOS

PASCAL: LOS ORIGENES - SACANDO CONEJOS DE LA MSX

Encienda una computadora Talent MSX y sus periféricos.

MODEM
- Interfaz asincrónica incorporada.
- Normas BELL y CCITT, 300 y 1.200 baudios.
- Incluye en 80 Kb software de comunicaciones, MSX-PLAN y MSX-WRITE.

UNIDAD DE DISCOS FLEXIBLES
- 5 1/4'' de 360 Kb (DS-DD)
- Velocidad transferencia 250 Kb/sg.
- Formato grabacion compatible MS-DOS.

MINI-LAN
EQUIPO CENTRAL PROFESOR — IMPRESORA
EQUIPO ALUMNO 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 — EQUIPO ALUMNO
DISKETTERAS 2 x 360 KB

(RED PARA USO EDUCACIONAL)
- Comunicación por línea compartida a 30.000 baudios.
- Comparte unidades de disco e impresoras de consola MSX maestro con hasta 10 consolas MSX alumno
- Carga simultánea de un programa a todos los alumnos.
- Carga individual de maestro a alumno
- Salvado de programa alumno a unidad de disco maestro.
- Salida a impresora de maestro del listado de programa alumno, en spooling.
- Funciona desde MSX-Basic, MSX-Logo o cualquier programa que corra desde Basic.
- Estando activa, se dispone de todo el MSX-Basic.

RS-232
- Velocidades programables desde 50 a 19.200 baudios.
- Procesamiento de comunicaciones desde Basic.
- Buffer de 128 caracteres para recepción

TECLADO NUMERICO
- Conexion a Joystick
- Se integra a todo el software que corre bajo MSX-DOS (Ej.: D-Base II, MSX-Plan, etc.)

MOUSE
- Código DPM-220, accesorio para graficar.

MONITOR MONOCROMATICO 12"
- Anti-reflejo - Fósforo verde.
- Apto para uso profesional.
- (80 caracteres x 25 líneas).
- Parlante con amplificador incorporado.

EXPANSION 80 COLUMNAS
- Hace posible la utilización de software estandar CP/M, emulando terminal tipo VT-52.
- Incluye software para manejo de video.

Software

MSX-LOGO	MSX-LPC	MSX-PLAN	MSX-WRITE
Desarrollado por Logo Computer System Inc. con aplicación de primitivas y redacción del Manual por los Ings. Hilario Fernández Long y Horacio Reggini.	Lenguaje de programación estructurado y en castellano.	Planilla de calculo de Microsoft Corp. (Versión para MSX del Multiplan.)	Procesador de palabra de ASCII Corp. en castellano.



Producida en San Luis por Telemática S.A. licenciataria exclusiva de Microsoft Corp. y ASCII Corp. para uso de la norma MSX en Argentina.
6 meses de garantía y mensualmente en su quiosco la revista Load MSX.

STAFF

LOAD MSX

Director

Fernando Flores

Secretario de Redacción

Ariel Testori

Redacción

Fernando Pedró

Arte y Diagramación

Boccardo Diseño Gráfico

Departamento de Publicidad

Dolores Urien

Servicios Fotográficos

Oscar Burriel y Claudio Veloso

EDITORIAL PROEDI

Presidente

Ernesto del Castillo

Vicepresidente

Cristian Pusso

Director Titular

Armengol Torres Sabaté

Director Suplente

Javier Campos Malbrán

Load Revista para usuarios de la norma MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5º Piso, (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Registro Nacional de la Propiedad Intelectual: E.T.M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de la Propiedad Intelectual.

ISSN 0326-8241. Impreso en Impresiones Gráficas Tabaré S.A.I.C., Erézcano 3158, Capital. Fotocromo de tapa: Columbia. Los ejemplares atrasados se venderán al precio del último número en circulación.

Distribuidor en Capital: Martino, Juan de Garay 358, P.B., Capital. Distribuidor en el interior: D.G.P., Hipólito Yrigoyen 1450, Capital, Tel.: 38-9266/9800.

CORREO ARGENTINO CASA CENTRAL — FRANQUEO PAGO CONCESION N — TARIFA REDUCIDA CONCESION N 1304

SUMARIO

Aplicación médica con MSX 2

Se trata del primer equipo de electroencefalografía computarizada basado en una "home computer" como es la MSX2.

El mercado de la electromedicina se caracterizó siempre, según informaron profesionales del medio, por tener que recurrir a equipos importados o bien nacionales con un alto costo de adquisición y mantenimiento, además de ser básicamente equipos de un sólo propósito y dedicado.

Con el advenimiento de la computación hogareña se hizo patente que la misma podría prestar una gran cantidad de servicios en esta rama, sobre todo para profesionales con inquietudes en el campo de la electromedicina.

Con motivo de las XXX Jornadas de Neurocirugía de la Provincia de Buenos Aires , realizadas en Pinamar del 8 al 12 de diciembre de 1988, Telemática S.A. - Talent MSX. y EMAC S.R.L. presentaron el equipo mencionado. A dicho congreso concurrieron neurocirujanos de todo el país y del exterior.

La Interfase CAD-3 permite presentar en pantalla las señales analógicas enviadas por un electroencefalógrafo EMAC EE-3, o bien de otra marca. Posibilita almacenar en memoria 10 minutos de sesión, con 8 canales separados.

A medida que se va almacenando la información, se puede detener la captura de datos y obtener una amplificación en tiempo de la señal almacenada, lo que permite ver con más detalle el electroencefalograma. Por pantalla se visualizan 8 segundos simultáneamente con los 8 canales; además se puede ampliar la señal para visualizar 4, 2 y 1 segundos respectivamente.

El software incorpora las facilidades de una pequeña base de datos (fichero) que permite guardar los datos del paciente, los de la toma y diversos comentarios que puedan servir como evaluación del electroencefalograma.

Por otra parte, el software integra la posibilidad de volcado por impresora con capacidad de gráficos (compatible MSX o Epson) de toda la sesión o de eventos seleccionables por el operador, en formato normal o ampliado.



Además, el operador tiene la posibilidad de copiar o descartar partes de la toma, lo que pone al alcance del profesional la potencia de un procesador de textos. Con la metodología empleada hasta el momento podría decirse que este equipo brinda las facilidades de un procesador de texto en un ámbito profesional habituado a trabajar con una máquina de escribir.

Finalmente, se puede realizar un histograma de todo el registro, así como un completo Análisis de Fourier del mismo.

Durante las Jornadas de Pinamar, el Doctor Peter Janetta, de Estados Unidos, manifestó que ellos poseen un equipamiento equivalente en su país basado en una Macintosh. Sin embargo, le pareció que este desarrollo íntegramente nacional es una solución inteligente para nuestro país y América Latina, donde la escasez de recursos para el área suele detener muchos proyectos.

EMAC S.R.L. ofrece la interfase con el software y el equipo completo, o sea computadora Talent MSX2 Turbo, una unidad de disquete Talent DPF-555, un monitor Talent y una impresora que puede ser Epson o cualquiera compatible, según desee el profesional.

Este equipamiento demuestra que la norma MSX brinda al mercado un sistema que por su modularidad facilita el desarrollo de sistemas de software y hardware para mercados verticales de gran especialización, y la Interfase CAD-3 es una opción más que interesante para el neurólogo.



Sin duda Chance ya ha sido consagrado como el juego más popular en todo el país.
Pero, detrás de todo gran éxito hay una muy buena razón. Por eso detrás de Chance, usted siempre encontrará a El Cronista Comercial.
Un éxito editorial que se escribe con la más completa información.
Desarrollada a través de un lenguaje claro y abierto.
Y con la perspectiva económica necesaria para comprender mejor la actualidad.
Hoy, quienes juegan con Chance, saben que la mejor información no es juego.

El Cronista Comercial

Su chance de asociarse al éxito.

LOS ORIGENES

El Pascal es un lenguaje de programación que fue creado para la enseñanza. Sin embargo, sus múltiples ventajas lo llevaron a situarse entre los favoritos de los programadores. Para ello fue decisivo el Turbo Pascal, que aportó dos tipos de mejoras: las palabras clave y los procedimientos incorporados.

Pascal fue inventado por Niklaus Wirth en el comienzo de la década del '70. (Su nombre es en honor de Blaise Pascal, un filósofo del siglo XVII). El lenguaje de programación Pascal se pensó inicialmente para la enseñanza -ayudaría a los programadores principiantes a desarrollar buenos hábitos al permitirles escribir programas claros, concisos y estructurados.

Antes del Pascal, lo que aprendían los estudiantes era FORTRAN, un lenguaje previo y no estructurado.

El profesor Wirth creyó que muchos errores comunes de programación podían evitarse si se utilizaba un lenguaje estructurado en bloques que fuera estricto en el tipo de variables utilizadas.

Pascal es un lenguaje de los que en inglés se suelen denominar "strongly typed", algo así como "fuertemente tipificado". Traducido a algo más comprensible, esto significa que las variables y constantes de un tipo no puede ser mezcladas libremente con variables y constantes de cualquier otro tipo. En otras palabras, a ambos lados de una asignación se deben colocar variables y constantes del mismo tipo. La única excepción aceptada es asignar a una variable real una expresión de tipo entero. La verificación que realizan los compiladores de Pascal sobre el tipo de variables utilizadas en una expresión permite evitar errores de programación al forzar al programador para que emplee variables compatibles.

En Turbo Pascal, los nombres de variables consisten en letras o subrayado (_) seguido por letras, dígitos o subrayados. La longitud máxima de un nombre de variable permitido es de 127 caracteres pero no es muy práctico poner nombres tales como:

Esta_variable_tiene_por_mision_contar

pues si tenemos que contar, la sentencia sería:

Esta_variable_tiene_por_mision_contar :=Esta_variable_tiene_por_mision_contar + 1;lo cual permite practicar mecanografía pero entorpece la programación.

Todos los strings en Turbo Pascal se encierran entre comillas simples ('). Esto difiere de lenguajes tales como el C o el BASIC donde se utilizan las comillas dobles (").

Pascal utiliza cinco tipos de datos escalares:



Palabra clave

char: Un carácter, longitud 1 byte

byte : Un entero, de un byte de longitud cuyo valor va de 0 a 255.

integer : Un entero con un valor de -32768 a 32767.

real.: Un número decimal con 11 dígitos de precisión

Boolean: TRUE o FALSE (verdad o falso).

Aunque a simple vista los tipos byte y char parecen ser lo mismo, la verificación estricta de tipo que hace Pascal impide utilizarlos indiscriminadamente. Las variables byte se usan como números enteros pequeños que aceleran la ejecución de un programa,

mientras que las tipo char son simples caracteres. Además de estos tipos de variables, Pascal permite generar arreglos de cada tipo escalar.
Aunque Pascal se desarrolló inicialmente para uso educativo, su facilidad de empleo y enfoque altamente estructurado ganaron muchos adeptos que habían programado en otros lenguajes. A medida que Pascal comenzó a ser utilizado para el desarrollo de software comercial, se necesitó extender el lenguaje que, entre otras cosas, no proveía de un buen manejo de archivos. Antes del Turbo Pascal, cada empresa de software que desarrollaba un compilador creaba su propio conjunto de extensiones, lo que dificultaba la portabilidad (esto es pasar un programa de una máquina a otra, o de una versión de un lenguaje a otro). Sin embargo, debido a que Turbo Pascal se convirtió en el estándar "de facto", muchas de las extensiones pueden ser utilizadas libremente hoy en día.
Respecto a la portabilidad, aquí surge una de las posibles aplicaciones, ya que hay una versión del Turbo Pascal en MSX (la 3.01 del CP/M-80) y es casi totalmente compatible con la existente en IBM PC y compatibles (previas a la 4.0 de PC). La transferencia entre una PC y una MSX de un programa escrito en Turbo Pascal es directa: basta tener en cuenta la extensión de la pantalla (80 columnas o 40 columnas) y recompilar el programa fuente (se puede cargar tranquilamente en cualquiera de las dos máquinas), que el programa en cuestión funciona sin problemas una vez realizado el "transplante". Otra de las ventajas del MSXDOS y van...

PASCAL COMO LENGUAJE ESTRUCTURADO

El lenguaje Pascal es estructurado y tiene ciertas similitudes con el Algol y el C. La característica distintiva de los lenguajes estructurados es su capacidad de armar bloques separados de datos y código. Esto significa que cada sección puede trabajar "por las suyas", ocultando toda la información del resto del programa para evitar incómodos efectos secundarios al efectuar una tarea específica. Solo se conocen los datos mínimos entre cada compartimiento y no se confunden entre sí. Cuando trabajamos en BASIC, más de una vez nos encontramos pensando si ya utilizamos esta variable o vemos cómo un ciclo termina de forma algo extraña y descubrimos que el índice del FOR..NEXT, por ejemplo, quedó modificado en la subrutina de la línea10000, llamada dentro del ciclo.
Generalmente, la compartamentalización se formaliza a través de subrutinas, comúnmente denominadas "subprogramas", con "variables locales", que son temporarias. De esta forma, es posible escribir subrutinas que no causen efectos secundarios en otras partes de un programa. El uso excesivo de las variables globales (variables "conocidas" a través de todo el programa) hará que los "bugs" trepen por nuestros programas al permitir la existencia de efectos secundarios (de por sí, lo más difícil de localizar cuando "debugueamos", o sea, buscamos errores). En Pascal, todos los subprogramas son funciones o procedimientos (procedure) discretos.
Las funciones y los procedimientos son los bloques constructivos de Pascal. Una tarea específica dentro de un programa puede ser definida y codificada en funciones o procedimientos. Luego de "debuguear" la función o procedimiento, un programador podrá confiar en la misma, ya que trabajará adecuadamente y sin efectos indeseables dentro del programa. Todas las variables declaradas en una función o procedimientos son reconocidas y admitidas únicamente por esa subrutina.
En Pascal, utilizar bloques de código crea lo que se llaman "estructuras de programa". Un "bloque de código" es un conjunto de sentencias Pascal conectadas lógicamente que pueden ser tratadas como una unidad. Se crean colocando líneas de código entre las palabras clave "begin" y "end". Por ejemplo, en BASIC escribiríamos:

```
100 IF X<10 THEN PRINT "Error. Intente de nuevo":X=-1:GOSUB 10000
```

donde 10000 podría ser una rutina que utiliza a X como bandera de algún proceso.
En Pascal queda mucho más visible la estructura.

```
if x < 10 then begin
WriteLn('Error. Intente de nuevo');
Anular(-1); end;
```

En este caso, la misma función realizada en BASIC se obtiene en Pascal en forma mucho más visible y, al utilizar una función, es innecesario tener disponible la variable X para la tarea de "mensajero", ya que se trabaja con una función de la misma manera que lo hacemos con las funciones estándares en BASIC: SIN, SQR, etcétera.
Vemos que el conjunto WriteLn.. Anular forma un bloque de programa que se ejecuta cuando se cumple la condición "x < 10" y puede ser pensado como una unidad -un bloque de código. Una sentencia no puede ejecutarse sin que se invoque la otra. En Pascal, cualquier sentencia puede ser única o un bloque de sentencias. El uso de los bloques de código genera programas legibles, con lógica fácil de seguir. Los bloques de código también permiten escribir programas mejores, libres de errores, porque el significado de cada sentencia es claro. Esto nos lleva a lo que se conoce como programación estructurada.

UN POCO DE PROGRAMACION ESTRUCTURADA

La forma en que la programación lineal o tradicional nos enseña a resolver problemas es tratar de obtener los algoritmos de cada procedimiento a

medida que vamos "creando" el programa. Esto nos lleva a estancarnos quizás en una rutina que muestra un mensaje de error y nos impide continuar con el programa, sin saber si al finalizar el mismo y verlo funcionando nos parecerá mejor utilizar otro tipo de algoritmo en una parte determinada del programa. Los "yanquis" denominan esta forma de programación "down-top" (de abajo hacia arriba), o sea de lo particular a lo general.
El enfoque estructurado se conoce también como "top-down" (de arriba hacia abajo), es decir de lo general a lo particular. Primero resolvemos el problema general, dejando para más adelante (una vez resuelto lo general) lo particular, los detalles. Este enfoque nos muestra cada procedimiento o conjunto de procedimientos como bloques estructurados, que se comportan como cajas negras que sabemos que hacen tal o cual tarea, pero no cómo la hacen.
Otra de las "máximas" de la programación estructurada es la abolición del GOTO. Nosotros, los curtidos programadores BASIC y Assembler, no podemos imaginarnos a veces qué sería la vida sin un GOTO salvador, pero... a veces los diagramas lineales de un programa quedan como la Avenida Libertador de nuestra Capital a las seis de la tarde: un nudo descomunal.
¿Cómo reemplazamos los GOTO? Muy fácil, con estructuras de control: FOR..NEXT, WHILE, REPEAT UNTIL, etcétera.
Todo esto nos lleva nuevamente al Pascal: si sumamos la posibilidad de utilizar subprogramas"estancos" ,estructuras de control que permiten obviar los GOTO y una forma de encarar la programación del tipo estructurado, nos encontraremos programando mucho más eficientemente, y, aunque no lo parezca, con mucha más rapidez.
Un programador Turbo Pascal puede crear y mantener una biblioteca personal de funciones que se ajusten a su personalidad. Ya que Turbo Pascal permite incluir distintos archivos-fuente dentro de un programa, los proyectos grandes se pueden manejar muy fácilmente.
A medida que el Pascal se fue haciendo más popular, muchos programadores comenzaron a utilizarlo en cualquier tarea (de programación, no les servía mucho para afeitarse, por ejemplo...). Ya que hay virtualmente al menos un compilador Pascal por cada computadora conocida (¡incluyendo la venerable TI99/4A!), se puede tomar un programa escrito en una máquina, compilarlo y correrlo en otra con pocos cambios, o ninguno. Esto ahorra mucho tiempo y dinero.
Veamos ahora qué nos trae el Turbo Pascal para los usuarios MSX.

MEJORAS INCORPORADAS AL TURBO PASCAL

En esta descripción sobre las mejoras del Turbo Pascal, presuponemos que el lector está familiarizado con el lenguaje y que conoce la versión estándar y sus limitaciones. Al que no lo conozca, igualmente le resultarán interesantes algunas consideraciones que se aplican a las generales de la programación.
La decisión de mejorar un lenguaje debe basarse en una buena razón. Cada extensión que se incorpore disminuye "per se" la portabilidad de los programas que la utilicen. Por otra parte, las extensiones hacen al lenguaje no estándar. Usualmente las extensiones se agregan porque la versión estándar del mismo carece de alguna función necesaria. En cuanto al Pascal estándar, las carencias que se observan son la falta de manejo de strings, direccionamiento absoluto, y manejo de overlays. Recordemos que el Pascal se creó como lenguaje de enseñanza y Borland, los creadores del Turbo Pascal, lo convirtieron en un lenguaje de propósito general, apto para el programador profesional.
Las mejoras introducidas por Borland caen en dos categorías: sentencias del lenguaje (palabras clave) y procedimientos incorporados. Veamos primero las sentencias.

a) Absolute

La primera mejora sustancial que encontramos en el Turbo Pascal es el modificador "absolute". El Pascal estándar no permite fijar un lugar de memoria determinado para una variable dada. Esta posibilidad era innecesaria en el área de aprendizaje de la programación, sin embargo, cuando se desarrolló software comercial o profesional, fue esencial.
Por ejemplo, supongamos que queremos modificar algún valor de la work-area del MSX más allá de la posición &HF380, digamos el que cuenta la cantidad de líneas de la pantalla (ver el truco que permite reservar unas líneas de la parte final de la misma).
En este caso, bastará con definir una variable de la siguiente forma:
lineas: byte absolute $F3B1
y luego, para poner 4 líneas reservadas, será suficiente con hacer la siguiente asignación:
lineas:=20;
Un ejemplo de aplicación de lo anterior es el siguiente programa:

```
program Lineas;
(*
  Este programa usa la variable del sistema
  CRTCNT para reservar las ultimas lineas
  aprovechando el 'absolute'
*)

var
  CrtCnt: byte absolute $F3B1;
  i: byte;
begin
   ClrScr;
   GotoXY(1,23);
   Write('Linea 24————————');
   GotoXY(1,21);
   Write('Linea 22————————');
   CrtCnt:=20;
   GotoXY(1,1);
   for i:=1 to 100 do
      WriteLn('i=',i:3);
```

```
  CrtCnt:=24;
end.
```

Nótese cómo se indican las variables en hexadecimal en Turbo Pascal. Siempre hay que recordar que el uso de variables absolutas debe hacerse con sumo cuidado para evitar que el sistema operativo y el programa choquen y "cuelguen" la computadora.
También se puede utilizar la sentencia absolute para hacer que dos variables compartan el mismo lugar de memoria. El siguiente ejemplo hace que las variables 'suma' y 'sigue' compartan la misma área:

```
suma : integer
sigue: integer absolute suma;
```

b) External

Con el Pascal estándar no es posible utilizar funciones o procedimientos que no estén escritos en Pascal. Sin embargo, hay muchas ocasiones en que es necesario que esto sea posible. Por ejemplo, si tenemos que controlar una interfase robótica que requiera alta velocidad de ejecución, lo más probable es que haya una rutina en código máquina que permita manejar el dispositivo.
En el Turbo Pascal para MSX, se permite declarar una subrutina escrita en código máquina a través de su dirección de memoria absoluta. Supongamos que utilizamos la clásica posición $D000 para almacenar nuestra rutina en Assembler. En este caso, la llamada a una supuesta rutina que maneje un interruptor sería:

```
procedure Enciende(cual: integer);
external $D000
```

El modificador 'external' le informa al Turbo Pascal que una rutina en Assembler será utilizada y en qué posición de memoria reside.
Se debe declarar un subprograma externo sin el cuerpo en nuestro programa Pascal; sólo es necesario especificar los parámetros del subprograma.



El problema que tiene external es que antes deberemos 'pokear' el subprograma en código máquina de alguna forma y analizar en qué posición de memoria lo colocamos para que no interfiera con el sistema operativo.
La forma en que se le pasan los parámetros a una rutina external en Turbo Pascal está más allá de esta nota, pero podríamos más adelante realizar algún ejemplo de aplicación. Para poder utilizar esta facilidad se debe conocer el lenguaje Assembler y la arquitectura de las MSX.

c) Inline

El Turbo Pascal permite insertar código máquina directamente en nuestro programa Pascal utilizando inline. Esta facilidad puede ser muy útil para realizar interfases con dispositivos especiales y escribir rutinas rápidas que empleen las facilidades del Turbo Pascal.
El código de máquina que se desea insertar en el programa se incluye a continuac ión de la palabra clave inline entre paréntesis. Cada byte o palabra se separa con la barra transversal (/). Se pueden utilizar los signos más (+) y menos (-) para sumar o restar. Un asterisco especifica una referencia al contador de posición dentro del programa Assembler. Como inline es una sentencia, termina con punto y coma (;). El siguiente ejemplo es una rutina que pasa a mayúsculas el string que se le envía. Al lado de los códigos se colocó como comentario el código fuente.

```
program Pinline;

(*
 Este programa utiliza la extension'inline'
 del Turbo Pascal para crear un procedimiento
 que pasa a mayusculas un string
*)

type
   Str = string [255];

var
   Alfa : Str;

procedure Mayusculas(var Strg: Str);

$A+

begin
    inline ($2A/Strg/  {   LD HL,(Strg)  }
            $46/       {   LD B,(HL)     }
            $04/       {   INC B         }
            $05/       {   L1: DEC B     }
            $CA/*+20/  {   JP Z,L2       }
            $23/       {   INC HL        }
            $7E/       {   LD  A,(HL)    }
            $FE/$61/   {   CP 'a'        }
            $DA/*-9/   {   JP C,L1       }
            $FE/$7B/   {   CP 'z'+1      }
            $D2/*-14/  {   JP  NC,L1     }
           $D6/$20/    {   SUB 20H       }
           $77/        {   LD  (HL),A    }
           $C3/*-20);  {   JP  L1        }
                       {L2: EQU $        }

end;

begin
   Write('Ingrese el string:');
   ReadLn(alfa);
   Mayusculas(alfa);
   WriteLn(alfa);
end.
```

d) Overlay

Una de las mejoras más importantes en el Turbo Pascal es la posibilidad del manejo de overlay. Este comando permite crear programas que son mayores que la capacidad de memoria de la máquina. Esto se hace automáticamente cargando subprogramas, llamados overlays, que se almacenan en disco hasta que se necesiten para la ejecución del programa.

Generalmente, para crear un overlay, se pone 'overlay' al principio de la función o "procedure" que se quiera usar. Cuando el Turbo Pascal encuentra el comando 'overlay', automáticamente compila ese subprograma en un archivo separado. Véase en el manual del Turbo Pascal la implementación de los overlays en este lenguaje.

e) shr y shl

Shr viene de 'shift right' y shl de 'shift left'. Estas operaciones pueden utilizarse sólo con variables del tipo "integer". Básicamente cumplen la misma función que los comandos de 'shifteo' que tiene el Z-80 con relación a los bytes.

f) String

Cuando utilizamos el Pascal estándar realmente no es muy fácil usar los strings como en, por ejemplo, el MSX BASIC. El Turbo Pascal agrega el tipo string, con funciones que permiten manipularlos con la misma facilidad que en BASIC. En oposición a los arreglos de caracteres, que deben tener una longitud fija, los string tienen longitud variable, que puede ir de 1 a la cantidad máxima especificada en la declaración de la variable (máximo 255 caracteres).

g) XOR

Turbo Pascal también agrega el operador XOR aplicable tanto a operandos del tipo "integer" como a los del tipo Boolean. Para los operandos" integer", se realiza la operación XOR entre los bits de cada "integer". Para los operandos Boolean, se obtiene TRUE o FALSE de acuerdo al resultado de la operación.

PROCEDIMIENTOS INCORPORADOS

Turbo Pascal incorpora un buen paquete de procedimientos que facilitan en gran medida la programación. Se pueden dividir en tres categorías: manejo de pantalla, strings y manejo del sistema operativo.

a) Procedimientos para manejo de pantalla

Estos son algunos de los procedimientos incorporados al Turbo Pascal:

Procedimiento	Efecto
CrtEol	Borra hasta el fin de la línea
CrtExit	Envía un string de reset a la pantalla
CrtInit	Envía un string de inicio a la pantalla
ClrScr	Borra pantalla
DelLine	Borra una línea
GotoXY	Equivale al LOCATE del BASIC
InsLine	Inserta una línea en blanco.

b) Funciones para strings

Como quedó dicho anteriormente, el Turbo Pascal incorpora el tipo 'string' y funciones para manejarlos. Algunos de estos procedimientos son:

Procedimiento	Efecto
Delete	Borra una parte de un string, dada la posición inicial y la cantidadde caracteres.
Insert	Inserta un string dentro de otro
Str	Convierte un integer o real a string
Val	Convierte un string a real o integer

Las siguientes funciones también soportan los strings:

Función	Efecto
Copy	Retorna un substring, copia de un string
Concat	Retorna un string que es el resultado de la concatenación de dos strings
Length	Equivale al LEN de BASIC
Pos	Equivale al INSTR de BASIC

c) Procedimientos para interfase con el sistema operativo

Turbo Pascal permite acceder al sistema operativo a través de dos procedimientos especiales: Bdos y Bios. En general, para que todo funcione correctamente en MSXDOS, deberemos utilizar el procedimiento Bdos, pues el Bios emplea parte del CP/M que es incompatible con el MSXDOS.

DIFERENCIAS ENTRE EL TURBO PASCAL Y EL PASCAL ESTANDAR

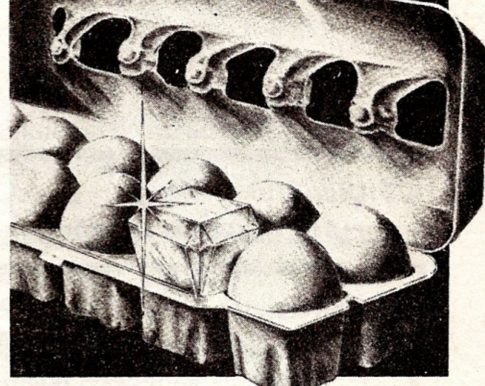
Aparte de las extensiones y mejoras incluidas por el Turbo Pascal, presenta algunas diferencias con el estándar. Los programadores experimentados de Pascal que desean utilizar el MSX con el Turbo deberán tenerlas en cuenta para evitar problemas.

a) El procedimiento 'new', que permite alojar variables dinámicas para utilizar con punteros, no acepta especificar un registro variante. Si se necesita trabajar con ellos, se deberá utilizar el procedimiento 'getmem'.

b) No se permite que un 'goto' salte fuera del bloque que se especifica. Esto implica que tanto el 'goto' como la etiqueta a donde se salta deben estar en el mismo bloque. (Pero si usamos programación estructurada, esto no es ningún problema... pues ¡nuestros programas no tienen 'goto'!)

c) No podremos utilizar parámetros tipo 'var' en rutinas recursivas (en la versión de PC esto está permitido).

d) No están implementadas 'get' ni



'put'; se debe usar 'read' y 'write' que fueron extendidas para su utilización con disquete.

e) Se puede utilizar la palabra clave 'packed', pero no tiene efecto. Asimismo,'pack' y 'unpack' no fueron implementados.

f) El procedimiento 'page' no está implementado.

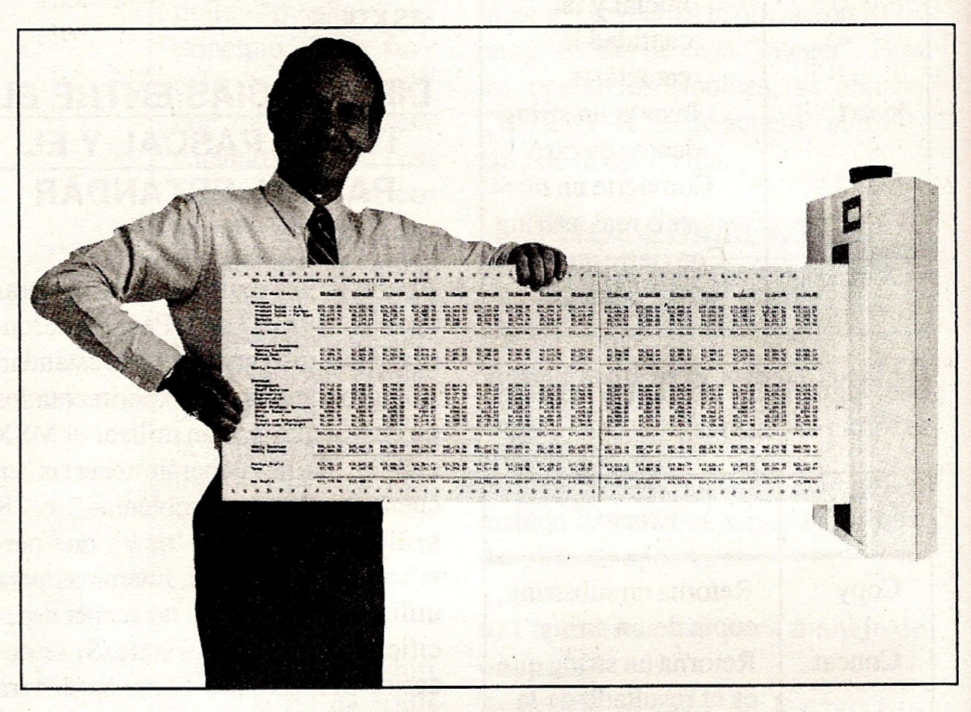
CONCLUSIONES

El MSXDOS permite acceder a la gran biblioteca de programas existente para CP/M-80, incluyendo famosos lenguajes como el Turbo Pascal. Viendo tanto su facilidad de uso como sus extensiones, es fácil ver cómo se convirtió en un estándar 'de facto' de los compiladores Pascal y cómo se ha extendido su uso a nivel profesional. (Como chimento final, diremos que la línea 'Agropack' -sistemas de aplicación para el agro que comercializa Telemática S.A.- fue desarrolladá en Turbo Pascal).

Hugo Daniel Caro

COMPUTO Y PRESUPUESTO

Tipo:Utilitario
Autor:Arq.Eduardo G.Armanino



Este programa es un utilitario destinado a arquitectos, ingenieros, constructores, etcétera que permite realizar el cómputo métrico y el presupuesto de una construcción a partir de cada una de sus partes, aunque con leves variantes podría ser utilizado por otras disciplinas.
Este programa esta dividido en dos partes, una la correspondiente al cómputo y la otra al presupuesto. En la primera se ingresan los datos correspondientes a cada uno de los items, es decir las dimensiones, y la computadora nos imprimirá el volumen o la superficie de los mismos, según lo que corresponda. En la segunda se ingresan los precios parciales y totales. Esta división permite que pueda realizarse, si así se desea, solamente el cómputo.
El manejo del programa es sencillo ya que en el menú aparecen las distintas operaciones posibles, y a medida que se va ejecutando se visualizan las instrucciones para ir continuando con el mismo.
La estructura general está marcada por esa división entre las dos partes, si bien en algunos sectores se mezclan las rutinas. Esto es debido a dos motivos: 1º) a que surgió como un programa simple, y luego se le fueron agregando mas operaciones y 2º) a que con la finalidad de aprovechar al máximo las rutina que se repiten en las distintas operaciones, hay bastante cantidad de 'gosub' y 'goto'.

INSTRUCCIONES:
Cómputo y presupuesto

Primeramente aparece en pantalla la presentación. En este punto se debe esperar unos instantes, ya que aquí se realizan los pokes correspondientes a las lecturas de la pantallas, se dimensionan las matrices, se redefinen los caracteres '$' y '?', y se coloca un ø a partir de la dirección 60000! hasta la dirección 61100!. Esto último es necesario debido a que las variables alfanúmericas no se ingresan ni se imprimen en pantalla como tales, sino que se van almacenando a partir de la dirección 60000! letra por letra, por su código ASCII. Esto permite ingresar todos los subítems que necesitemos sin alterar la memoria disponible para cadena de caracteres. Se pueden ingresar hasta diez subitems de seis caracteres cada uno, por cada uno de los dieciocho items. Esta cantidad de items puede ser variada, de acuerdo a las necesidades de cada uno teniendo las siguientes precauciones: 1º) dimensionar las matrices en la línea 200 reemplazando el Nº 18 por el que corresponda, 2º) cambiar el bucle de la línea 210 con el valor que corresponda y 3º) cambiar el bucle de la línea 500 con el mismo valor y agregar o eliminar los datos de las líneas 510 y 520. Se debe tener la precaùcuión de no agregar demasiada cantidad de items ya que se puede agotar la memoria disponible.
Luego de la presentación se inicia con el cómputo métrico. Oprimiendo cualquier tecla se visualiza el menú. Oprimiendo la letra 'K' se inicia desde el ítem 0. En cada item se pueden ingresar hasta diez subitems de seis caracteres cada uno. Estos pueden ser mayúsculas o minúsculas o números. Si algún subitem tuviera menos de seis caracteres, los faltantes se deben completar con espacios. Una vez ingresados los seis carácteres se pasa directamente a ingresar las dimensiones, sin oprimir 'enter'. El oprimir 'enter' mientras estamos ingresando el nombre de algún subitem ocasionará que borre el mismo y nos permita realizar correcciones en caso de haber ingresado un dato erróneo, también nos permite pasar al item siguiente o abandonar el cómputo.
Para cada subitem se pueden ingresar hasta tres dimesiones: largo, ancho y alto. Si fuera un subitem que no tiene

dimensión sino que se mide por unidades se debe colocar la cantidad de las mismas en la primera dimensión y 0 en las otras dos. Una vez ingreadas las dimensiones de un subitem la computadora imprimirá la superficie, volumen, unidades del mismo, según lo que corresponda.
La primer dimensión de un subitem siempre debe ser mayor que cero, ya que de lo contrario ocurre lo mismo que al oprimir 'enter' mientras ingresabamos el nombre del subitem.
En el caso de que un item tenga menos de diez subitems, una vez ingresado el último necesario, se oprimen las teclas 'graph' + 'p' y se pasa automáticamente al item siguiente.
Una vez finalizado el cómputo métrico se pueden visualizar en el menú las distintas operaciones que se pueden realizar, y que son las siguientes:
1) Listar los nombres de los items. Oprimiendo 'L' aparecerá toda la lista.
2) Listar un item. Oprimiendo 'I' aparecerá el item correspondiente con todos los datos ingresados hasta el momento.
3) Modificar un item. Oprimiendo 'A' se podrá agregar o quitar cualquier subitem. Primero la computadora preguntará si se desea agregar o quitar, debiéndose ingresar 'a' o 'd' según el caso. Luego preguntará a que item corresponde la modificación y lo listará, preguntando por el subitem a agregar o quitar. En el caso de que algún subitem ya existente se desee modificar parcialmente, primero se debe anular y luego se debe agregar nuevamente.
4) Realizar otro cómputo. Oprimir 'O'.
5) Finalizar. Oprimir 'F'.
6) Pasar al presupuesto. Oprimir'P'.
En el presupuesto se inicia de forma similar al cómputo, apareciendo primeramente todas las opciones disponibles.
Oprimiendo 'K' se comienza con el primer item. Se visualizarán en la pantalla todos los subitems correspondientes con sus correspondientes superficies, volúmenes o unidades, según corresponda. La visualización de los subitems se irá haciendo uno por uno, debiéndos ingresar el precio unitario de cada uno y la computadora imprimirá el precio parcial. Así se debe realizar con todos los subitems hasta llegar al último. Si algún item tiene menos de diez subitems basta con oprimir 'graph' + 'p' y se imprimirá en la pantalla el precio total del item y se pasará al siguiente.
Una vez finalizado el presupuesto ocurre lo mismo que con el cómputo, es decir que aparecen todas las opciones.
Una vez realizado el presupuesto cualquier modificación de algún item que se quiera realizar, se deberá hacer tanto en el cómputo como en el presupuesto, tarea que la computadora nos indicará automáticamente.
Para listar un item, igual que en el caso anterior, el listado se realizará del item indicado en el cómputo y en el presupuesto.
En el caso de modificar algún item, ya sea agregando o quitando algún subitem, se modificará automáticamente el precio total del item, imprimiéndose el mismo en la pantalla. También se modificará en este caso el precio total de la obra.
En el presupuesto también tenemos la opción 'C' que nos permite visualizar en pantalla el nombre de todos los item con sus correspondientes precios y el porcentaje que le corresponde a cada uno en el total de la obra y se imprimirá en la parte inferior de la pantalla el costo total de la misma.

ESTRUCTURA DEL PROGRAMA:

10-240: Preparación: dimensionamiento de vectores y mátrices, redefinición de caracteres, pokes.
250-390: Preparación cómputo métrico.
400-490: Elección de opciones.
500-540: Lista de ítems.
550-1040: Inicio de cómputo métrico.
1050-1410: Modificación de item: agregar o eliminar subitem.
1420-1710: Lista un item completo (cómputo solamente).
1720-1810: Preparación presupuesto.
1820-1910: Elección de opciones.
1920-2280: Inicio de presupuesto.
2290-2420: Costo total de la obra.
2430-2650: Modificación de item (cómputo y presupuesto).
2660-2840: Lista de un item completo (con presupuesto).
2850-3030: Rutinas que se repiten en varias operaciones.

LISTA DE VARIABLES:

I = Nº de item.
G = Nº de subitem.
VC = Código ASCII del carácter ingresado.
NN = Dirección de memoria donde se almacena VC.
PE = Peek de una dirección de memoria.
CA (I,G) = Dimensión a de un subitem.
CB (I,G) = '' b '' '''''
CC (I,G) ='' c '' '''''
continúa la lista de variables de Cómputo y Presupuesto.
VD (I,G) = Volumen, superficie o cantidad de un subitem.
PP (I,G) = Precio unitario de un subitem.
V (I,G) = Precio total de un subitem.
PIT (I) = Precio total de un item.
FIN = Precio total de la obra
M = Posición para imprimir en pantalla.
B$ (I) = Nombre de un ítem.
Z$ = Opción elegida.

```
0   'computo y presupuesto
2 '1989 eduardo g. armanino
5 'preparacion
10 KEYOFF:SCREEN0:WIDTH40:VDP(7)
=49
20 CLEAR200
30 CLS:E$=">>>>>>>>>>>>>>>>>>>><
<<<<<<<<<<<<<<<<<<":PRINTE$:LOCA
TE0,21:PRINTE$
40 LOCATE10,10:PRINT"un minuto p
or favor"
50 PRINT:PRINT:PRINT"C O M P U T
 O  y  P R E S U P U E S T O"
60 T=BASE(2)+8*63:FORH=TTOT+7
70 READA:VPOKEH,A:NEXT
80 T=BASE(2)+8*36:FORH=TTOT+7
90 READA
100 VPOKEH,A:NEXT
110 DATA128,192,224,240,240,224,
192,128,32,32,112,248,80,248,136
,0
120 'lectura de pantallas
130 DEFUSR8=51770!:DEFUSR9=51810
!
140 FORR=51750!TO51839!
150 READD:POKER,D:NEXTR
160 DATA64,68,72,76,80,84,88,92,
96,100,104,108,112,116,120,124,0
,0,0,0
170 DATA1,12,0,33,0,0,62,32,205,
86,0,33,38,202,58,248,247,133,11
1,30,0,86,33,0,0,1,192
180 DATA3,62,84,211,168,205,89,0
,62,80,211,168,201,33,38,202,58,
248,247,133,111,126,103,46,0,17,
0,0,1,192,3,62,84,211,168,205,92
,0,62,80,211,168,201
190 FORI=60000!TO61100!:POKEI,0:
NEXT
200 DIMB$(18),V(18,10),CA(18,10)
,CB(18,10),PIT(18),CC(18,10),VD(
18,10),PP(18,10),PO(18)
210 FORI=0TO17:READB$(I):NEXT
220 C$="descrip.      a       b
      c       v"
230 D$="presione cualquier tecla
 para continuar"
240 W$="
                  ":U=16:CLS
250 LOCATE3,5:PRINT" ***   **  *
  * ***  *  * ***  **       *
 *  * ** ** *  * *  *  *  *  *
    *    *  * * * * ***  *  *  *
  *  *"
260 LOCATE3,8:PRINT"*      *  * *
  * *    *  *  *  *  *      ***
 **  *    * *    **   *   **"
270 LOCATE0,16:PRINTD$
280 IFINKEY$=""THEN280
290 CLS:A$="--------------------
--------------------":PRINT
300 PRINT"M E N U":PRINTA$
310 PRINT"K   comienza":PRINT
320 PRINT"L   lista items":PRINT
330 PRINT"A   agrega o desagrega
 item":PRINT
340 PRINT"I   lista un item":PRI
NT
350 PRINT"F   finaliza":PRINT
360 PRINT"O   otro c"mputo":PRIN
T
370 PRINT"P   presupuesta":PRINT
380 PRINT"C   costo total y porc
entajes"
390 LOCATE0,22:PRINTE$:EC=USR8(0
)
400 Z$=INKEY$:IFZ$=""THEN400
410 IFZ$="K"THEN500
420 IFZ$="I"THEN1430
430 IFZ$="L"THEN500
440 IFZ$="P"THEN1730
450 IFZ$="A"THEN1060
460 IFZ$="O"THEN20
470 IFZ$="F"THENEND
480 IFZ$="M"THENLEC=USR9(0)
490 GOTO400
500 CLS:FORI=0TO17:PRINTB$(I):NE
XT
510 DATA0  derechos y honorarios
,1  trabajos preliminares,2 mam
posteria,3  hormigon y losas cer
am.,4  aislaciones,5  cubiertas,
6  revoques,7  cielorrasos
520 DATA8  contrapisos,9  pisos
y zocalos,10 revestimientos,11 c
arpinteria,12 inst. sanitarias,1
3 inst. gas,14 inst. electricas,
15 pintura,16 vidrios,17 varios
530 IFZ$="K"THEN550
540 GOTO1570
550 LOCATE0,20:PRINTD$,E$
560 IFINKEY$=""THEN560
570 FORI=0TO17:GOSUB1690
580 GOSUB2950
590 FORG=0TO9
600 'escribir subitem
610 LOCATE0,U:PRINTD$
620 K$=INKEY$:IFK$=""THEN620
630 LOCATE0,U:PRINTW$
640 IFK$=CHR$(230)THENGOSUB1020:
GOTO610
650 IFK$=CHR$(148)THENLC=USR9(0)
:GOTO 400
660 IFK$=CHR$(219)THEN980
670 VPOKEM,G+48
680 M=M+1:BEEP:VPOKEM,63
690 FORD=1TO6:M=M+1:VPOKEM,219
700 V$=INKEY$:IFV$=""THEN700
710 IFV$=CHR$(219)THEN960
720 IFV$<>CHR$(13)THEN750
730 M=M-D-1:FORW=1TO6:VPOKEW+M,0
:NEXT
740 FORW=NN-(D-2)TONN:POKEW,0:NE
XT:LOCATE0,G+5:PRINTW$:GOTO 610
750 VC=ASC(V$):VPOKEM,VC
760 GOSUB3020:POKENN,VC:BEEP:NEX
TD:VPOKEM-6,0
770 LOCATE10,G+5:BEEP:INPUTCA(I,
G)
780 LOCATE10,G+5:PRINTSPC(9)
790 LOCATE10,G+5:PRINTUSING"###.
##";CA(I,G)
800 IFCA(I,G)=0THEN1550
810 'ingreso b
820 LOCATE17,G+5:BEEP:INPUTCB(I,
G)
830 LOCATE17,G+5:PRINTSPC(9)
840 LOCATE17,G+5:PRINTUSING"####
#.##";CB(I,G)
850 IFCB(I,G)=0THENVD(I,G)=CA(I,
G):GOTO 930
860 'ingreso c
870 LOCATE25,G+5:BEEP:INPUTCC(I,
G)
880 LOCATE25,G+5:PRINTSPC(10)
890 LOCATE25,G+5:PRINTUSING"####
#.##";CC(I,G)
900 IFCC(I,G)=0THENVD(I,G)=CA(I,
G)*CB(I,G):GOTO 930
910 'v subitem
920 VD(I,G)=CA(I,G)*CB(I,G)*CC(I
,G)
930 LOCATE33,G+5:PRINTUSING"####
.##";VD(I,G)
940 M=M+33
950 IFZ$="A"THENU=16:GOTO 1570
960 NEXTG
970 IFINKEY$=""THEN970
980 BEEP:NEXTI
990 IFINKEY$=""THEN990
1000 BEEP:LC=USR9(0):GOTO400
1010 'escrit pantalla actual
1020 EC=USR8(1):LC=USR9(0)
1030 FORJ=1TO1500:NEXT:LC=USR9(1
):RETURN
1040 GOTO 400
1050 'modif item
1060 CLS:PRINTA$:U=18
1070 BEEP:INPUT"agrega o desagre
ga (a/d)";AG$
1080 IFAG$="a"THENBG$="agregar":
GOTO 1110
1090 IFAG$="d"THENBG$="desagrega
r":GOTO 1110
1100 CLS:GOTO 1070
1110 PRINT:BEEP:PRINT"ingrese el
 n#mero de item a modificar":INP
UTI
1120 IFI>17THEN1060ELSECLS:PRINT
:PRINTB$(I):PRINTA$;
1130 PRINTC$,A$
1140 M=200
1150 FORG=0TO9
1160 VPOKEM,G+48
1170 FORD=1TO6:GOSUB3030
1180 IFPE=0THEN1210
1190 VPOKEM+D+1,PE:NEXT
1200 GOSUB1640
1210 M=M+40:NEXTG
1220 LOCATE0,15:PRINTA$;
1230 BEEP:LOCATE0,20:PRINT"ingre
se n#mero de subitem a ";BG$:BEE
P:INPUTG
1240 IFG>9THENLOCATE0,20:PRINTW$
:GOTO 1230
1250 IFG>9THEN1230ELSEMG=G
1260 IFAG$="d"THEN1310
1270 'agregar subitem
1280 LOCATE0,20:PRINTW$,E$
1290 M=(G+5)*40:GOTO610
1300 'desagregar subitem
1310 LOCATE0,20:PRINTW$,E$
1320 FORD=1TO6
1330 GOSUB3020:POKENN,0:BEEP
1340 NEXTD
1350 CA(I,G)=0:CB(I,G)=0:CC(I,G)
=0:VD(I,G)=0
1360 LOCATE0,16:PRINTD$:LOCATE1,
G+5:PRINTW$
1370 IFINKEY$=""THEN1370
1380 IFPR<1THENCLS:LC=USR9(0):GO
TO 400
1390 PIT(I)=PIT(I)-V(I,G):FIN=FI
N-V(I,G)
1400 IFINKEY$=""THEN1400
1410 GOTO 2670
1420 'lista un item
1430 CLS
1440 PRINT:BEEP:INPUT"que item q
uiere listar";I
1450 IFI>17THEN1430ELSEM=200
1460 GOSUB1690
1470 FORG=0TO9:VPOKEM,G+48
1480 FORD=1TO6:GOSUB3030
1490 IFPE=0THEN1520
1500 VPOKEM+D+1,PE:NEXT
1510 GOSUB1640
1520 M=M+40:NEXTG
1530 LOCATE0,15:PRINTA$;
1540 GOTO 1570
1550 LOCATE0,G+5:PRINTW$
1560 M=M-7:GOTO 610
1570 LOCATE0,21:PRINTD$
1580 IFINKEY$=""THEN1580
1590 IFZ$="I"ANDPR>0THEN2670
1600 IFZ$="A"ANDPR>0THEN2440
1610 LC=USR9(0)
1620 IFPR>0THEN1820
1630 GOTO 400
1640 LOCATE10,G+5:PRINTUSING"###
.##";CA(I,G)
1650 LOCATE19,G+5:PRINTUSING"###
.##";CB(I,G)
1660 LOCATE27,G+5:PRINTUSING"###
.##";CC(I,G)
1670 LOCATE33,G+5:PRINTUSING"###
```

```
#.##";VD(I,G)
1680 RETURN
1690 CLS:PRINT:BEEP:PRINTB$(I):P
RINTA$;
1700 PRINTC$,A$
1710 RETURN
1720 'inicio presup
1730 CLS:PR=1
1740 LOCATE6,2:PRINT"***  ***  *
***  *** *  *                *
* *  * *    *    *  *
     ***  ***  ***   **  *  *
             *    *  * *       *
 *  *                *    *  * *
*** ***   **"
1750 LOCATE4,9:PRINT"***    *  * *
***  *** ***  **             *
* *  * *    *     *  *  *
     ***  *  * ***   **    *  *
*            *    *  * *       *
  *  *  *            *     **  *
*** ***   *   **"
1760 LOCATE0,19:PRINTD$
1770 S$="tipo               cant.
 $/u        $total"
1780 U=16
1790 F$="precio total del item:
$"
1800 IFINKEY$=""THEN1800
1810 LEC=USR9(0)
1820 Z$=INKEY$:IFZ$=""THEN1820
1830 IFZ$="K"THENCLS:GOTO 1920
1840 IFZ$="I"THEN1430
1850 IFZ$="A"THEN1060
1860 IFZ$="O"THEN20
1870 IFZ$="L"THEN500
1880 IFZ$="C"THEN2300
1890 IFZ$="F"THENEND
1900 IFZ$="M"THENLC=USR9(0)
1910 GOTO1820
1920 FORI=0TO17:GOSUB2920
1930 GOSUB2950
1940 FORG=0TO9
1950 'escribir subitem
1960 LOCATE0,U:PRINTD$
1970 K$=INKEY$:IFK$=""THEN1970
1980 LOCATE0,U:PRINTW$
1990 IFK$=CHR$(230)THENGOSUB2270
:GOTO 1960
2000 IFK$=CHR$(148)THEN2840
2010 IFK$=CHR$(219)THEN2180
2020 VPOKEM,G+48
2030 FORD=1TO6:GOSUB3030
2040 IFPE=0THEN2150
2050 VPOKEM+D+1,PE:NEXT
2060 'cant subitem
2070 LOCATE13,G+5:PRINTUSING"###
.##";VD(I,G)
2080 IFVD(I,G)=0THEN2800
2090 'precio unit
2100 LOCATE20,G+5:BEEP:INPUTPP(I
,G)
2110 LOCATE20,G+5:PRINTSPC(10)
2120 LOCATE20,G+5:PRINTUSING"###
##.##";PP(I,G)
2130 'precio subitem
2140 V(I,G)=VD(I,G)*PP(I,G):LOCA
TE30,G+5:PRINTUSING"######.##";V
(I,G)
2150 M=M+40
2160 IFZ$="A"THEN2620
2170 NEXTG
2180 FORN=0TOG:PIT(I)=PIT(I)+V(I
,N):NEXT
2190 LOCATE0,15:PRINTA$;:GOSUB28
90
2200 LOCATE0,22:PRINTD$
2210 IFINKEY$=""THEN2210
2220 FIN=FIN+PIT(I)
2230 NEXTI
2240 IFINKEY$=""THEN2240
2250 BEEP:LEC=USR9(0):GOTO1820
2260 'escrit pantalla actual
2270 EC=USR8(1):LC=USR9(0)
2280 FORJ=1TO1500:NEXT:LC=USR9(1
):RETURN
2290 'costo total
2300 CLS
2310 BEEP:PRINT"descripci”n
       $       %"
2320 FORI=0TO17
2330 BEEP:PRINTB$(I)
2340 IFPIT(I)=0THEN2370
2350 PO(I)=PIT(I)/FIN*100
2360 LOCATE22,I+1:PRINTUSING"###
###.##";PIT(I);:PRINTTAB(34)USIN
G"##.##";PO(I)
2370 NEXT
2380 PRINT"costo total de obra:"
;:PRINTTAB(22)USING"######.##";F
IN;:PRINTTAB(33)"100.00",A$
2390 LOCATE0,21:PRINTD$,E$
2400 IFINKEY$=""THEN2400
2410 LC=USR9(0)
2420 GOTO 1820
2430 'modif item
2440 CLS:PRINT:PRINTB$(I):PRINTA
$;
2450 PRINTS$,A$
2460 M=200
2470 FORG=0TO9
2480 VPOKEM,G+48
2490 FORD=1TO6:GOSUB3030
2500 IFPE=0THEN2530
2510 VPOKEM+D+1,PE:NEXT
2520 GOSUB2850
2530 M=M+40:NEXTG
2540 LOCATE0,15:PRINTA$;
2550 BEEP:PRINTF$;:PRINTTAB(30)U
SING"######.##";PIT(I)
2560 FIN=FIN-PIT(I):PIT(I)=PIT(I
)-V(I,G)
2570 'modif subitem
2580 LOCATE0,20:PRINTW$,E$
2590 G=MG
2600 LOCATE20,G+5:PRINTSPC(19)
2610 M=(G+5)*40:GOTO2100
2620 PIT(I)=PIT(I)+V(I,G)
2630 GOSUB2890
2640 FIN=FIN+PIT(I)
2650 U=16:GOTO 2820
2660 'lista un item
2670 CLS
2680 M=200
2690 GOSUB2920
2700 FORG=0TO9
2710 VPOKEM,G+48
2720 FORD=1TO6:GOSUB3030
2730 IFPE=0THEN2760
2740 VPOKEM+D+1,PE:NEXT
2750 GOSUB2850
2760 M=M+40:NEXTG
2770 LOCATE0,15:PRINTA$;
2780 GOSUB2890
2790 GOTO 2820
2800 LOCATE0,G+5:PRINTW$
2810 GOTO 1960
2820 LOCATE0,21:PRINTD$
2830 IFINKEY$=""THEN2830
2840 LC=USR9(0):GOTO 1820
2850 LOCATE13,G+5:PRINTUSING"###
.##";VD(I,G)
2860 LOCATE20,G+5:PRINTUSING"###
##.##";PP(I,G)
2870 LOCATE30,G+5:PRINTUSING"###
###.##";V(I,G)
2880 RETURN
2890 BEEP:LOCATE0,16:PRINTF$
2900 LOCATE30,16:PRINTUSING"####
##.##";PIT(I)
2910 RETURN
2920 CLS:PRINT:BEEP:PRINTB$(I):P
RINTA$;
2930 PRINTS$,A$
2940 RETURN
2950 LOCATE0,17:PRINTA$
2960 LOCATE0,18:PRINT"code + f
=  finaliza comp. o presup."
2970 PRINT"enter        =  reingres
a subitem"
2980 PRINT"code + m  =  men#"
2990 PRINT"graph + p =  contin#a
 al item siguiente"
3000 BEEP:PRINTE$
3010 M=200:RETURN
3020 NN=60000!+I*60+G*6+D-1:RETU
RN
3030 PE=PEEK(60000!+I*60+G*6+D-1
):RETURN
```

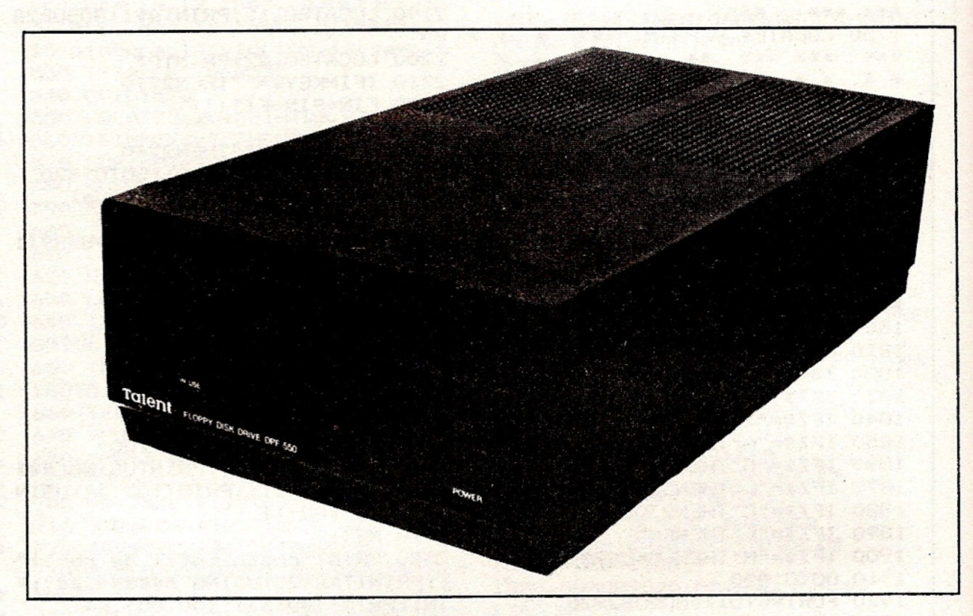
PERIFERICOS TALENT

No todos los usuarios conocen realmente la variedad de periféricos que existen para su MSX. Más aún,desconocen las aplicaciones que pueden tener,como por ejemplo administrar una báscula para camiones de 60 toneladas.

En general, lo primero que observa el usuario potencial de una computadora es la gama de periféricos que puede disponer para su posible adquisición.
En el caso de las Talent MSX, la gama es amplia y está pensada para un conjunto muy interesante de aplicaciones dentro de las computadoras hogareñas. Demás está decir que como son equipos désarrollados para la norma MSX, cualquier equipo incluído en la misma podrá utilizar estos periféricos sin problemas.

a) Unidad de discos

El primer gran paso que suele dar el usuario novato es comprar una unidad de discos, ya que luego de trabajar con un datasete, las ventajas en cuanto a confiabilidad y velocidad son importantes.



En el caso de las unidades Talent, la DPF-550/555 es la que actualmente se encuentra disponible en plaza y permite almacenar 360 kbytes en disquetes de 5 1/4 pulgadas. Su velocidad de transferencia permite que la carga y grabación de datos sea muy rápida y confiable. Además es compatible en lectura y grabación con las PC vía el MS-DOS lo que permite intercambiar datos entre esas computadoras y nuestra MSX.
Por otra parte, si somos felices poseedor de una MSX2 Turbo, podrá utilizar sin modificaciones el CP/M Plus, otro sistema operativo más veloz que funciona en estas computadoras.

CENTROS DE ASISTENCIA AL USUARIO DE TALENT MSX

CAPITAL FEDERAL

Taller Logo de computación
Junín 1930

Centro de Computación Clínica
Asistencia al Usuario Discapacitado
Ramsay 2250 - Pabellón F
Tel. 784-2018

Barrio Norte
Uriburu 1063 - Tel. 83-6892/826-6692

Belgrano
Cabildo 735 - Tel. 772-9088

Flores
Gral. Artigas 354 - Tel. 612-3902

Palermo
Guatemala 4733 - Tel. 71-4124

San Telmo
Chile 1345 - Tel. 37-0051 al 54

GRAN BUENOS AIRES

Caseros
Medina y Alberdi - 1er piso - Tel. 750-8016.

Castelar
C. Casares 997 - Tel. 629-2247

Lanús
Caaguazú 2186 - Tel. 240-0378

Morón
Belgrano 160 - Tel. 629-3347

Ramos Mejía
Bolívar 55 - 1er. piso - Tel. 658-4777

San Justo
Perú 2515 (esq. Arieta)

Vicente López
Av. Maipú 625 - Tel. 797-672

Virreyes - Pdo. de San Fernando
Avellaneda 1697 - Tel. 745-7963

b) Monitor

La segunda adquisición suele ser el monitor monocromático, sobre todo si el usuario tiene intención de usar su computadora en aplicaciones "profesionales", es decir, en contabilidad, ingeniería y ciencias. La necesidad del monitor surge de que, aunque los colores que nos brinda la computadora son muy bonitos, para trabajar durante un tiempo medianamente prolongado no son muy recomendables para la vista, pues generalmente producen fatiga ocular.

c) Modem

En los últimos dos años hemos visto surgir el furor por la telemática, que es la conjunción de la informática con las telecomunicaciones. En nuestro país comenzaron a surgir diversas bases de datos y mensajerías electrónicas que hacen de este periférico el punto central de atención.
Gracias al mismo, un usuario de La Quiaca podrá enviar sus datos en forma instantánea a otro ubicado en Ushuaia, si ambos poseen modem (y teléfono, por supuesto).

El ámbito de aplicación de la telemática se va extendiendo día a día y cada vez son más los usuarios que piensan que este periférico debe formar parte de su equipamiento.

d) Teclado numérico

Cuando hablamos de aplicaciones "serias" comúnmente nos viene a la mente la utilización de una Talent MSX en el ámbito de la pequeña y mediana empresa, donde los sistemas de contabilidad, sueldos y jornales, stock, gestión de ventas, etcétera campean sin problemas.
El usuario que utiliza estas aplicaciones suele ser un administrativo que, acostumbrado a usar calculadoras de tira de papel, realiza sus ingresos numéricos en forma más eficiente si posee un teclado similar al de estos equipos.
De allí la utilidad del teclado numérico en la administración y en planillas de cálculo.
Nota: En general la pregunta más realizada respecto al teclado numérico es si es posible utilizarlo con el MSX Plan o el Multiplan. La respuesta es no y sí. No se puede usar con el MSX Plan por una limitación que este software posee respecto a instalar rutinas externas al mismo. Y sí, pues el Multiplan trabaja en el MSXDOS y las rutinas implementadas en él funcionan sin problemas, ya sea en Multiplan, Wordstar,Calcstar, MBASIC, FORTRAN, COBOL, Turbo Pascal, etcétera.

e) Interfase RS-232C

En general, se desconoce la aplicación que puede tener este periférico en el entorno MSX (o en cualquier computadora). Su principal utilidad es la de brindar un vínculo estándar con otros periféricos o cualquier dispositivo que lo incorpore, incluyendo obviamente las computadoras.
Entre otras posibilidades, una RS232C permite interconectar su MSX a una impresora, a un modem (tipo Hayes compatible, que son los que utilizan las computadoras PC), equipos de distinto calibre (por ejemplo BaireSoft implementó un sistema para administrar una báscula para camiones de 60 tn a una empresa contratista de SOMISA),etcétera.
Realmente la mayor utilidad que se le puede dar a una RS232C con la MSX es trabajar como terminal de una computadora (mayor). Por ejemplo, con una PC AT trabajando en Xenix, es muy fácil utilizar un software "emulador de terminal". La MSX se comporta como un teclado más de dicha computadora, con la ventaja de ser una computadora en sí misma. En plaza no se ha aprovechado mucho esta variante, pero realmente es algo

muy interesante y económico para las grandes instalaciones.

f) Mouse

Es uno de esos periféricos que están dirigidos a un mercado más específico: aquellos que utilizan sus computadoras en aplicaciones gráficas o en un entorno gráfico al estilo Apple Macintosh.
Para los que se mueven en este ámbito, un periférico como el mouse de MSX es imprescindible para trabajar con soltura. Es uno de los que los yanquis denominan "pointing device" (dispositivo puntero) y tiene aplicación para dibujantes, grafistas, autoedición, arquitectos, etcétera.

g) MiniLAN

En realidad la MiniLAN II configura un conjunto de periféricos que unidos permiten un mejor aprovechamiento de una instalación informática, generalmente en el ámbito educativo.
Dados los costos actuales del hardware, no es nada despreciable la posibilidad de que hasta 10 consolas puedan compartir una o dos disqueteras y, opcionalmente, una impresora, con muy pocas diferencias respecto a la utilización directa de estos periféricos.

En una nota posterior nos extenderemos sobre estos periféricos educacionales, que permiten el acceso de todos los alumnos a aplicaciones del MSXDOS y MSX BASIC o LOGO, incluyendo compiladores de otros lenguajes como C, Turbo Pascal, COBOL y FORTRAN.

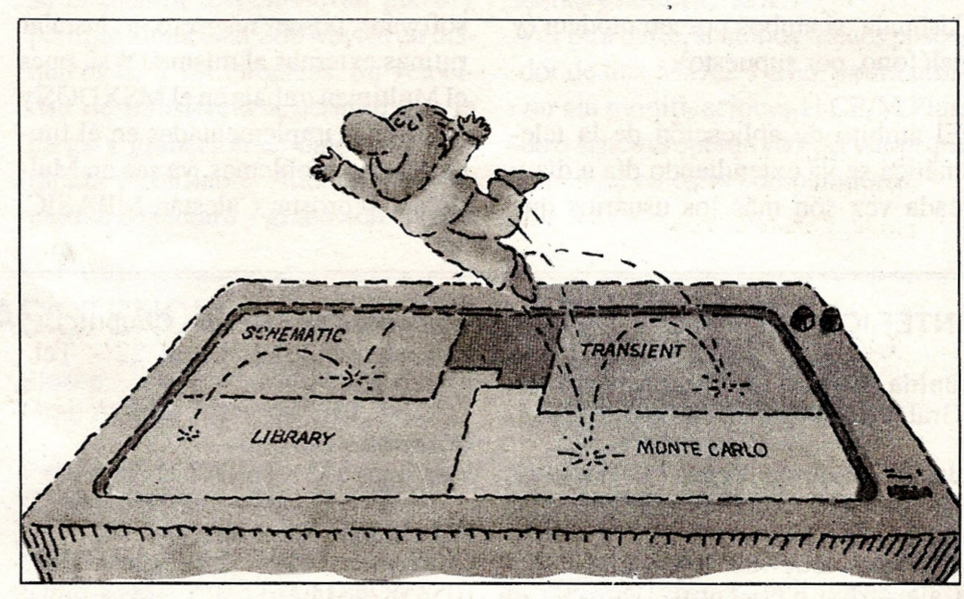
TALENT MSX EN MAR DEL HOP

Gracias a un acuerdo celebrado entre Telemática S.A. y la producción del programa Mar del Hop! que conduce Raúl Portal, los equipos Talent MSX vuelven a los sets de televisión.

En un esfuerzo importante se han facilitado en préstamo cerca de 9 equipos completos para su uso en el programa.

El encargado de computación que colabora con Raúl Portal, Osvaldo Espósito, nos comentó que en el primer tramo del programa solo se utilizaron nuestros equipos en dos tareas: el juego "Kenio", que no es ni más ni menos que el viejo y querido Simon, y el seguimiento de los números del "Mambo", versión portalesca del Bingo que utiliza modelos femeninos como bolillas (¡!).

Para cuando estas líneas ganen la calle se incorporará el uso de la digitalizadora, que ya se encuentra en venta para el público. Se tomarán "fotos digitalizadas" de los artistas que suelen visitar el programa y luego se les entregará una copia impresa autografiada.



Por último está previsto ampliar el "Kenio" con una interfase para manejar siete luces de 220v en un tablero que incorporará pulsadores.

No es la primera vez que Telemática S.A. colabora con la televisión. Hemos estado en programas como "La hora de los juegos" en Canal 11, "Magazine" con Silvia Martínez y Luisa Delfino a principios del año pasado en ATC, "Onda Joven" en Canal 11, y "Todo al Nueve" en Canal 9, en donde las pantallas de preguntas se manejan con equipos MSX.

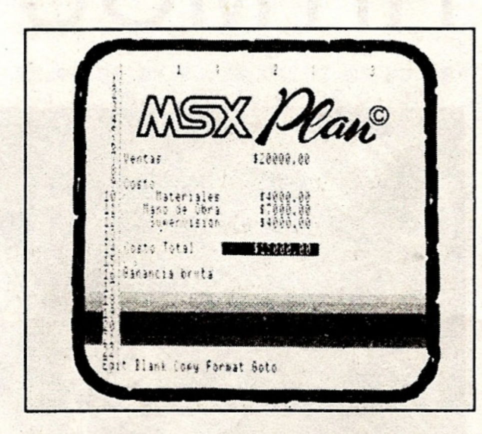
CONTESTANDO A LA HOT LINE

Planillas

¿Cuál es la diferencia entre el MSX Plan y el Multiplan para MSX? He visto que son muy similares y que pueden intercambiar planillas.

Respuesta

La diferencia principal es que el MSX Plan es un subconjunto del Multiplan y tiene algunas extensiones que permiten utilizarlo con casete. Por otra parte, el MSX Plan viene en cartucho, mientras que el Multiplan se consigue en disquete. Algunas de las posibilidades que incorpora el Multiplan,y que no vienen con el MSX Plan, son el uso de planillas externas (external), ancho variable de las columnas, posibilidad de uso ventanas para dividir la pantalla, ayuda en línea, etcétera. Por otra parte, el MSX Plan es comercializado por Telemática S.A., lo que implica la existencia de manual en castellano, cursos en los Centros de Asistencia, etcétera.
Como conclusión, si ya tiene experiencia con el Multiplan y posee disquetera, lo más conveniente es usar el Multiplan de MSX. En cambio si no tiene disquetera,o no tiene experiencia en el uso de planillas de cálculo, la mejor opción es usar el MSX Plan,ya que permite el empleo de casetes y su manejo es mucho más sencillo que el del Multiplan.



Procesadores de texto

¿Cuáles son los procesadores de texto existentes en el mercado para MSX?

Respuesta

Los procesadores de texto existentes en el mercado son los siguientes:

MSX1 y 2

MSX Write
Idea Text
MS Text
Tasword
Wordstar (con 80 columnas)

MSX 2

Wordstar
Tasword II
Melbrane Notes
Philips Home Office 1 (paquete que incluye un procesador de texto)
Philips Home Office 2 (paquete que incluye un procesador de texto)
Philips Ease (paquete que incluye un procesador de texto)y algunos más .

Cobol

¿Qué compiladores COBOL hay disponibles para MSX? ¿Viene alguno en cartucho?

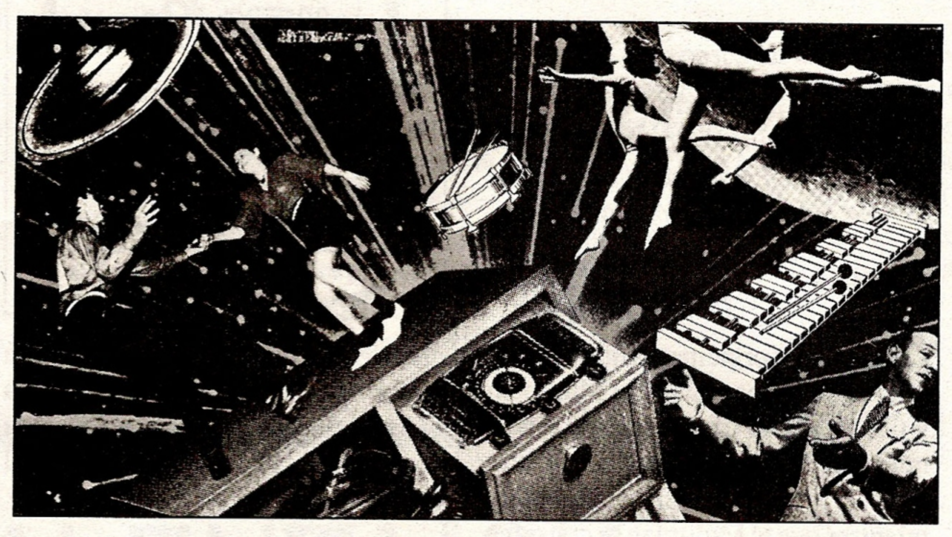
Respuesta

Los compiladores COBOL disponibles para MSX vienen en disquete únicamente y son 3 (los que conocemos):

RM COBOL
Microsoft COBOL 80
Nevada COBOL (sólo en MSX2 - CP/M Plus)

RITMOS

CLASE: Utilitario
AUTOR: Gonzalo Rodríguez Mir
Participó en el 4º Concurso de programas



El programa simula el sonido de una batería; puede programar dos ritmos distintos a la vez, con sus ondas de sonido, y combinarlos.

Inicialmente las ondas están programadas simulando un redoblante, un bombo y un charleston (representados respectivamente por los dibujos), sin embargo las ondas de volumen y tono del redoblante y del charleston pueden programarse a voluntad, pues son exactamente iguales en su tratamiento. El sonido de bombo no puede ser modificado y es el único realizado con el tono del PSG, los otros dos están hechos con ruido.

La variedad de sonidos que se logra programando las ondas se puede apreciar mucho mejor conectando la salida de audio de la computadora a un equipo de alta fidelidad.

El programa puede ser útil para aquellos músicos que deseen tocar o realizar grabaciones con este tipo de acompañamiento. También puede resultar interesante para quienes quieran aprender a dominar los tiempos en música.

El programa carece de cualquier tipo de efectos sonoros y también se ha desactivado el sonido "clik" de pulsación de teclado debido a que esto resulta molesto cuando se trabaja con volumen muy alto.

FUNCIONAMIENTO

En primer lugar se seleccionan las longitudes de ambos ritmos con las cuatro teclas de cursor. Si se quiere hacer un ritmo de 3x4, se escogerá 12, para uno de 2x4 se pondrá 8 o 16, etcétera. Una vez elegidos, se presiona la barra espaciadora y se pasa a la pantalla principal.

En esta pantalla se dibuja la estructura del ritmo deseado moviendo el cursor en las cuatro direcciones y oprimiendo la barra para imprimir o borrar un punto en el lugar correspondiente.

Presionando la tecla "ESC", el cursor pasa a la parte derecha de la pantalla. Una vez ahí, si se mueve hacia arriba o abajo, el cursor se desplaza, y hacia la izquierda o derecha, se modifican los números de repetición, velocidad del ritmo y velocidad de las ondas.

A la derecha de cada ritmo figura la cantidad de repeticiones del ritmo correspondiente, y más abajo la cantidad general. de repeticiones.

VELOCIDAD RITMO: Cuanto menor es el número, mayor es la velocidad con que se ejecuta.

VELOCIDAD ONDA: Similar a los registros 11 y 12 del comando **SOUND**. Cuanto menor es el número, más se "acorta" el sonido del golpe.

Volviendo a presionar la tecla "ESC", el cursor vuelve a la parte de dibujo de los ritmos.

Teclas de función:

• F1- EJEC: Se realiza automáticamente una introducción de 3 golpes para marcar la entrada y a continuación se ejecuta el ritmo, que puede ser interrumpido con **CTRL+STOP**.

•F2- ONDAS: Se visualizan las ondas de sonido. La de arriba es la de volumen, la de abajo de tono, y corresponden al sonido indicado por el dibujo. Con las teclas hacia la izquierda y derecha se desplaza el cursor. Hacia arriba y abajo se modifica la onda en el punto correspondiente. Con la barra espaciadora se pasa a la onda siguiente o vuelve a la pantalla principal.

•F3- INICIO: Vuelve al inicio (se borran el ritmo y las ondas que hay en memoria).

•F4- SAVE: Se almacena en casete un ritmo con sus valores y ondas. Primero se ingresa el nombre (hasta 6 caracteres) y luego se pulsa **RETURN**.

•F5- LOAD: Carga un ritmo previamente grabado en casete.

CARGADOR BASIC

Se carga con **CLOAD**, después de teclear **RUN**, el programa tarda un minuto aproximadamente hasta terminar su ejecución.

Para ejecutar el programa en código máquina se debe teclear: **DEFUSR= &HDØØØ: X= USR (Ø).**

El listado del cargador **basic** tiene al principio,para quienes quieran te-

Continúa en pág. **22**

```
10 FOR A=0 TO 51
20 S=0:DI=&HD000+A*82:DF=DI+81
30 FOR N=DI TO DF:READ A$:POKE N
,VAL("&H"+A$):S=S+PEEK(N)
40 NEXT
50 READ T$
60 IF S<>VAL("&H"+T$) THEN BEEP:
PRINT " ERROR EN LINEA";(A*2+8)*
10;"O";(A*2+9)*10
70 NEXT
80 DATA 3E,00,32,DB,F3,CD,CC,00,
0E,01,06,E2,CD,47,00,3E,01,32,7F
,F8,3E,00,32,80,F8,3E,05,32,8F,F
8,3E,00,32,90,F8,3E,02,32,9F,F8,
3E,00,32,A0,F8,3E,06,32,AF,F8,3E
,00,32,B0,F8,3E,04,32,BF,F8,3E,0
0,32,C0,F8,21,00,38,3E,00,CD,4D,
00,23,3E,00,CD,4D,00,23,3E,00
90 DATA 1DCC
100 DATA CD,4D,00,23,3E,00,CD,4D
,00,23,3E,0F,CD,4D,00,23,3E,30,C
D,4D,00,23,3E,C0,CD,4D,00,23,3E,
B0,CD,4D,00,23,3E,AF,CD,4D,00,23
,3E,A1,CD,4D,00,23,3E,A1,CD,4D,0
0,23,3E,81,CD,4D,00,23,3E,80,CD,
4D,00,23,3E,70,CD,4D,00,23,3E,0F
,CD,4D,00,23,3E,00,CD,4D,00,23
110 DATA 199C
120 DATA 3E,00,CD,4D,00,23,3E,00
,CD,4D,00,23,3E,00,CD,4D,00,23,3
E,00,CD,4D,00,23,3E,E0,CD,4D,00,
23,3E,18,CD,4D,00,23,3E,06,CD,4D
,00,23,3E,1A,CD,4D,00,23,3E,EA,C
D,4D,00,23,3E,0A,CD,4D,00,23,3E,
0A,CD,4D,00,23,3E,02,CD,4D,00,23
,3E,02,CD,4D,00,23,3E,1C,CD,4D
130 DATA 16CD
140 DATA 00,23,3E,E0,CD,4D,00,23
,3E,00,CD,4D,00,23,3E,00,CD,4D,0
0,23,3E,01,CD,4D,00,23,3E,0F,CD,
4D,00,23,3E,10,CD,4D,00,23,3E,20
,CD,4D,00,23,3E,40,CD,4D,00,23,3
E,40,CD,4D,00,23,3E,80,CD,4D,00,
23,3E,80,CD,4D,00,23,3E,80,CD,4D
,00,23,3E,80,CD,4D,00,23,3E,40
150 DATA 1780
160 DATA CD,4D,00,23,3E,40,CD,4D
,00,23,3E,20,CD,4D,00,23,3E,10,C
D,4D,00,23,3E,0F,CD,4D,00,23,3E,
00,CD,4D,00,23,3E,C0,CD,4D,00,23
,3E,20,CD,4D,00,23,3E,90,CD,4D,0
0,23,3E,48,CD,4D,00,23,3E,1C,CD,
4D,00,23,3E,24,CD,4D,00,23,3E,14
,CD,4D,00,23,3E,14,CD,4D,00,23
170 DATA 171B
180 DATA 3E,14,CD,4D,00,23,3E,14
,CD,4D,00,23,3E,28,CD,4D,00,23,3
E,28,CD,4D,00,23,3E,50,CD,4D,00,
23,3E,A0,CD,4D,00,23,3E,00,CD,4D
,00,23,3E,00,CD,4D,00,23,3E,00,C
D,4D,00,23,3E,01,CD,4D,00,23,3E,
01,CD,4D,00,23,3E,07,CD,4D,00,23
,3E,18,CD,4D,00,23,3E,FF,CD,4D
190 DATA 171F
200 DATA 00,23,3E,30,CD,4D,00,23
,3E,0F,CD,4D,00,23,3E,03,CD,4D,0
0,23,3E,01,CD,4D,00,23,3E,01,CD,
4D,00,23,3E,01,CD,4D,00,23,3E,01
,CD,4D,00,23,3E,01,CD,4D,00,23,3
E,01,CD,4D,00,23,23,23,23,23,3E,
C0,CD,4D,00,23,3E,30,CD,4D,00,23
,3E,FE,CD,4D,00,23,3E,18,CD,4D
210 DATA 1619
220 DATA 00,23,3E,E0,CD,4D,00,23
,3E,80,CD,4D,00,21,60,38,3E,C6,C
D,4D,00,23,3E,82,CD,4D,00,21,66,
38,3E,82,CD,4D,00,23,3E,C6,CD,4D
,00,21,E0,38,3E,18,CD,4D,00,23,3
E,24,CD,4D,00,23,3E,42,CD,4D,00,
3E,0F,32,E9,F3,3E,04,32,EA,F3,3E
,04,32,EB,F3,CD,62,00,3E,01,32
230 DATA 1CA7
240 DATA AF,FC,CD,5F,00,06,07,3E
,D1,21,00,1B,CD,4D,00,23,23,23,2
3,10,F7,21,00,C0,06,60,23,36,C8,
10,FB,26,01,2E,08,CD,C6,00,21,11
,DA,CD,0A,DA,26,01,2E,0A,CD,C6,0
0,21,29,DA,CD,0A,DA,3E,08,32,00,
C0,32,FF,BF,3E,48,32,62,C0,3A,FF
,BF,CD,C7,D9,3E,38,32,62,C0,3A
250 DATA 1F2E
260 DATA 00,C0,CD,C7,D9,21,00,1B
,3E,37,CD,4D,00,23,3E,C4,CD,4D,0
0,23,3E,0C,CD,4D,00,23,3E,01,CD,
4D,00,3E,38,32,62,C0,21,00,1B,3E
,37,CD,4D,00,3E,02,32,AF,FC,CD,9
F,00,F5,3E,01,32,AF,FC,F1,1E,20,
BB,CA,B1,D3,1E,1C,BB,28,0C,1E,1D
,BB,28,18,1E,1F,BB,28,27,18,CD
270 DATA 1D2C
280 DATA 3A,FF,BF,1E,10,BB,28,C5
,3C,32,FF,BF,CD,C7,D9,18,BC,3A,F
F,BF,1E,02,BB,28,B4,3D,32,FF,BF,
3A,FF,BF,CD,C7,D9,18,A8,3E,48,32
,62,C0,21,00,1B,3E,47,CD,4D,00,3
E,02,32,AF,FC,CD,9F,00,F5,3E,01,
32,AF,FC,F1,1E,20,BB,28,37,1E,1C
,BB,28,0D,1E,1D,BB,28,19,1E,1E
290 DATA 2371
300 DATA BB,CA,01,D3,18,CD,3A,00
,C0,1E,10,BB,28,C5,3C,32,00,C0,C
D,C7,D9,18,BC,3A,00,C0,1E,02,BB,
28,B4,3D,32,00,C0,3A,00,C0,CD,C7
,D9,18,A8,3E,01,32,64,C0,32,EB,C
0,3E,32,32,66,C0,32,EC,C0,3E,00,
32,65,C0,21,67,C0,06,0F,70,23,10
,FC,06,11,36,00,23,10,FB,06,20
310 DATA 1FCE
320 DATA 36,17,23,10,FB,06,08,70
,23,10,FC,06,18,36,00,23,10,FB,0
6,20,36,1E,23,10,FB,3E,0F,32,E9,
F3,CD,62,00,3E,01,32,AF,FC,CD,5F
,00,21,40,06,3E,10,CD,4D,00,23,3
E,10,CD,4D,00,23,3E,10,CD,4D,00,
23,3E,10,CD,4D,00,23,3E,10,CD,4D
,00,23,3E,10,CD,4D,00,23,3E,10
330 DATA 171B
340 DATA CD,4D,00,23,3E,10,CD,4D
,00,23,3E,10,CD,4D,00,23,3E,10,C
D,4D,00,23,3E,38,CD,4D,00,23,3E,
7C,CD,4D,00,23,3E,7C,CD,4D,00,23
,3E,38,CD,4D,00,23,3E,10,CD,4D,0
0,23,3E,10,CD,4D,00,21,19,20,3E,
1F,CD,4D,00,21,00,1B,3E,1A,CD,4D
,00,23,3E,0A,CD,4D,00,23,3E,00
350 DATA 158A
360 DATA CD,4D,00,23,3E,0F,CD,4D
,00,23,3E,2A,CD,4D,00,23,3E,0A,C
D,4D,00,23,3E,04,CD,4D,00,23,3E,
0F,CD,4D,00,23,3E,3A,CD,4D,00,23
,3E,0A,CD,4D,00,23,3E,08,CD,4D,0
0,23,3E,0F,CD,4D,00,23,3E,52,CD,
4D,00,23,3E,0A,CD,4D,00,23,3E,00
,CD,4D,00,23,3E,0F,CD,4D,00,23
370 DATA 1598
380 DATA 3E,62,CD,4D,00,23,3E,0A
,CD,4D,00,23,3E,04,CD,4D,00,23,3
E,0F,CD,4D,00,23,3E,72,CD,4D,00,
23,3E,0A,CD,4D,00,23,3E,08,CD,4D
,00,23,3E,0F,CD,4D,00,23,3E,1F,C
D,4D,00,23,3E,20,CD,4D,00,23,3E,
0C,CD,4D,00,23,3E,01,CD,4D,00,3A
,FF,BF,CB,27,CB,27,CB,27,C6,18
390 DATA 18AB
400 DATA 32,63,C0,3A,00,C0,CB,27
,CB,27,CB,27,C6,18,32,ED,C0,26,0
3,2E,05,CD,C6,00,3A,FF,BF,47,21,
00,C0,23,7E,DF,10,FB,26,03,2E,07
,CD,C6,00,3A,FF,BF,47,21,10,C0,2
3,7E,DF,10,FB,26,03,2E,09,CD,C6,
00,3A,FF,BF,47,21,20,C0,23,7E,DF
,10,FB,26,03,2E,0C,CD,C6,00,3A
410 DATA 2116
420 DATA 00,C0,47,21,30,C0,23,7E
,DF,10,FB,26,03,2E,0E,CD,C6,00,3
A,00,C0,47,21,40,C0,23,7E,DF,10,
FB,26,03,2E,10,CD,C6,00,3A,00,C0
,47,21,50,C0,23,7E,DF,10,FB,21,D
6,18,3E,C8,CD,4D,00,21,B6,19,3E,
C8,CD,4D,00,21,36,1A,3E,C8,CD,4D
,00,21,76,1A,3E,C8,CD,4D,00,21
430 DATA 1E41
440 DATA B6,1A,3E,C8,CD,4D,00,26
,14,2E,03,CD,C6,00,21,41,DA,CD,0
A,DA,26,01,2E,12,CD,C6,00,21,4A,
DA,CD,0A,DA,26,01,2E,14,CD,C6,00
,21,6D,DA,CD,0A,DA,26,01,2E,16,C
D,C6,00,21,5A,DA,CD,0A,DA,26,01,
2E,13,CD,C6,00,21,81,DA,CD,0A,DA
,26,01,2E,15,CD,C6,00,21,89,DA
450 DATA 1F8D
460 DATA CD,0A,DA,3E,30,32,62,C0
,3A,64,C0,CD,C7,D9,3E,68,32,62,C
0,3A,65,C0,CD,C7,D9,3E,88,32,62,
C0,3A,EB,C0,CD,C7,D9,3E,98,32,62
,C0,3A,66,C0,CD,C7,D9,3E,A8,32,6
2,C0,3A,EC,C0,CD,C7,D9,3E,20,32,
61,C0,3E,20,32,62,C0,3E,07,1E,FF
,CD,93,00,3E,02,32,AF,FC,CD,9F
470 DATA 29A2
480 DATA 00,F5,3E,01,32,AF,FC,F1
,67,3E,06,BC,CA,BA,DF,3E,04,BC,C
A,3E,E0,3E,01,BC,CA,B0,DD,3E,05,
BC,CA,91,DA,3E,02,BC,CA,7B,D2,3E
,20,BC,C2,21,D7,3A,61,C0,47,3E,9
8,B8,38,BC,21,00,00,3A,62,C0,6F,
CB,25,CB,14,CB,25,CB,14,E5,01,00
,00,3A,61,C0,4F,CB,39,CB,39,CB
490 DATA 2630
500 DATA 39,E1,09,01,00,18,09,CD
,4A,00,47,3E,C8,B8,28,04,3E,C8,1
8,02,3E,C9,F5,CD,4D,00,3A,62,C0,
5F,3E,20,BB,20,05,21,01,C0,18,2B
,3E,30,BB,20,05,21,11,C0,18,21,3
E,40,BB,20,05,21,21,C0,18,17,3E,
58,BB,20,05,21,31,C0,18,0D,3E,68
,BB,20,05,21,41,C0,18,03,21,51
510 DATA 17CB
520 DATA C0,3A,61,C0,CB,3F,CB,3F
,CB,3F,D6,04,4F,06,00,09,F1,77,C
3,5A,D6,3E,1B,BC,20,21,3A,61,C0,
47,3E,B0,B8,28,0D,3E,B0,32,61,C0
,3E,30,32,62,C0,C3,4E,D8,3E,20,3
2,61,C0,32,62,C0,C3,4E,D8,3A,61,
C0,47,3E,B0,B8,CA,62,D8,3E,1E,BC
,20,4C,3A,62,C0,47,3E,20,B8,20
530 DATA 2322
540 DATA 19,3A,61,C0,47,3A,ED,C0
,B8,D2,70,D7,3A,ED,C0,32,61,C0,3
E,78,32,62,C0,C3,4E,D8,3E,58,B8,
20,19,3A,61,C0,47,3A,63,C0,B8,D2
,8E,D7,3A,63,C0,32,61,C0,3E,40,3
2,62,C0,C3,4E,D8,3A,62,C0,0E,10,
91,32,62,C0,C3,4E,D8,3E,1F,BC,20
,49,3A,62,C0,47,3E,78,B8,20,19
550 DATA 2594
560 DATA 3A,61,C0,47,3A,63,C0,B8
,D2,C1,D7,3A,63,C0,32,61,C0,3E,2
0,32,62,C0,C3,4E,D8,3E,40,B8,20,
18,3A,61,C0,47,3A,ED,C0,B8,D2,DF
,D7,3A,ED,C0,32,61,C0,3E,58,32,6
2,C0,18,68,3A,62,C0,C6,10,32,62,
C0,18,5E,3E,1D,BC,20,2A,3A,61,C0
,47,3E,20,B8,20,15,3A,62,C0,1E
570 DATA 240B
```

clearlo, una rutina que detecta en qué número de línea hay error .
Es importante grabarlo antes de ejecutarlo. El programa en código máquina se graba con **BSAVE, "CAS: RITMOS", &HDØØØ, &HEØ98**, y se carga con **BLOAD "CAS:", R.**

```
580 DATA 4B,BB,30,05,3A,63,C0,18
,03,3A,ED,C0,32,61,C0,18,3B,3A,6
1,C0,0E,08,91,32,61,C0,C3,4E,D8,
3E,1C,BC,C2,5A,D6,3A,61,C0,47,3A
,62,C0,1E,4B,BB,30,05,21,63,C0,1
8,03,21,ED,C0,7E,B8,20,07,3E,20,
32,61,C0,18,0A,3A,61,C0,C6,08,32
,61,C0,18,00,21,18,1B,3A,62,C0
590 DATA 1E7E
600 DATA 3D,CD,4D,00,23,3A,61,C0
,CD,4D,00,C3,5A,D6,3E,1E,BC,20,3
0,3A,62,C0,47,3E,30,B8,20,07,3E,
A8,32,62,C0,18,D7,3A,62,C0,1E,68
,BB,20,04,0E,38,18,0B,1E,88,BB,2
0,04,0E,20,18,02,0E,10,3A,62,C0,
99,32,62,C0,18,B7,3E,1F,BC,20,30
,3A,62,C0,47,3E,A8,B8,20,07,3E
610 DATA 1B9F
620 DATA 30,32,62,C0,18,A2,3A,62
,C0,1E,30,BB,20,04,0E,38,18,0B,1
E,68,BB,20,04,0E,20,18,02,0E,10,
3A,62,C0,81,32,62,C0,18,82,3E,1C
,BC,20,78,3A,62,C0,1E,30,BB,28,1
1,1E,68,BB,28,1F,1E,88,BB,28,2D,
1E,98,BB,28,3B,18,4C,3A,64,C0,06
,63,B8,CA,5A,D6,3C,32,64,C0,CD
630 DATA 1B60
640 DATA C7,D9,C3,5A,D6,3A,65,C0
,06,63,B8,CA,5A,D6,3C,32,65,C0,C
D,C7,D9,C3,5A,D6,3A,EB,C0,06,63,
B8,CA,5A,D6,3C,32,EB,C0,CD,C7,D9
,C3,5A,D6,3A,66,C0,06,63,B8,CA,5
A,D6,3C,32,66,C0,CD,C7,D9,C3,5A,
D6,3A,EC,C0,06,63,B8,CA,5A,D6,3C
,32,EC,C0,CD,C7,D9,C3,5A,D6,3E
650 DATA 2FA1
660 DATA 1D,BC,C2,5A,D6,3A,62,C0
,1E,30,BB,28,11,1E,68,BB,28,1F,1
E,88,BB,28,2D,1E,98,BB,28,3B,18,
4C,3A,64,C0,06,01,B8,CA,5A,D6,3D
,32,64,C0,CD,C7,D9,C3,5A,D6,3A,6
5,C0,06,00,B8,CA,5A,D6,3D,32,65,
C0,CD,C7,D9,C3,5A,D6,3A,EB,C0,06
,01,B8,CA,5A,D6,3D,32,EB,C0,CD
670 DATA 2679
680 DATA C7,D9,C3,5A,D6,3A,66,C0
,06,01,B8,CA,5A,D6,3D,32,66,C0,C
D,C7,D9,C3,5A,D6,3A,EC,C0,06,01,
B8,CA,5A,D6,3D,32,EC,C0,CD,C7,D9
,C3,5A,D6,F5,3A,62,C0,CB,3F,CB,3
F,CB,3F,3C,6F,26,19,CD,C6,00,F1,
6F,3E,00,26,00,0E,0A,06,08,CB,25
,17,B9,F5,3F,CB,14,F1,38,01,91
690 DATA 281F
700 DATA 10,F2,F5,3E,30,84,DF,3A
,62,C0,CB,3F,CB,3F,CB,3F,3C,6F,2
6,1A,CD,C6,00,F1,C6,30,DF,C9,7E,
23,DF,B7,20,FA,C9,4C,4F,4E,47,49
,54,55,44,20,43,4F,4D,50,41,53,2
D,52,49,54,4D,4F,31,20,00,4C,4F,
4E,47,49,54,55,44,20,43,4F,4D,50
,41,53,2D,52,49,54,4D,4F,32,20
710 DATA 1F0E
720 DATA 00,52,45,50,45,54,49,43
,2E,00,46,31,2D,45,6A,65,63,20,2
0,20,54,4F,54,41,4C,00,46,35,2D,
4C,6F,61,64,20,20,20,56,45,4C,2E
,4F,4E,44,41,00,46,33,2D,49,6E,6
9,63,20,20,20,56,45,4C,2E,52,49,
54,4D,4F,00,46,32,2D,4F,6E,64,61
,00,46,34,2D,53,61,76,65,00,06
730 DATA 1484
740 DATA 07,3E,D1,21,00,1B,CD,4D
,00,23,23,23,23,10,F7,3E,01,32,E
7,C0,3E,01,32,E9,F3,CD,82,00,3E,
02,32,AF,FC,CD,5F,00,CD,C3,00,21
,00,1B,3E,D1,CD,4D,00,3E,07,32,F
2,F3,21,FF,FF,23,E5,11,45,00,19,
44,4D,11,00,00,CD,11,01,CD,20,01
,06,BD,CD,08,01,CD,20,01,10,F8
750 DATA 1CC1
760 DATA E1,3E,01,BD,20,DF,11,40
,00,01,45,00,CD,11,01,CD,20,01,0
6,84,CD,FC,00,CD,20,01,10,F8,21,
FF,FF,11,BD,00,01,45,00,CD,11,01
,CD,20,01,06,84,CD,FC,00,CD,20,0
1,10,F8,3E,47,32,61,C0,3E,30,32,
62,C0,21,1C,1B,3E,3F,CD,4D,00,23
,3E,46,CD,4D,00,23,3E,1C,CD,4D
770 DATA 1BC0
780 DATA 00,23,3E,0F,CD,4D,00,21
,03,1B,3A,E7,C0,1E,01,BB,28,03,2
1,0B,1B,3E,0F,CD,4D,00,2B,2B,3E,
20,CD,4D,00,2B,3E,32,CD,4D,00,3E
,1C,32,B7,FC,3E,1C,32,B9,FC,3E,5
6,CD,8D,00,3E,4F,CD,8D,00,3E,4C,
CD,8D,00,3E,2E,CD,8D,00,3E,1C,32
,B7,FC,3E,6E,32,B9,FC,3E,54,CD
790 DATA 1C10
800 DATA 8D,00,3E,4F,CD,8D,00,3E
,4E,CD,8D,00,3E,4F,CD,8D,00,21,8
7,C0,3A,E7,C0,1E,01,BB,28,03,21,
A7,C0,06,20,C5,7E,E5,21,20,00,48
,06,00,ED,42,CB,25,CB,25,11,48,0
0,19,CB,27,CB,27,2F,CE,3D,44,4D,
5F,16,00,CD,2A,DD,E1,23,C1,10,D9
,06,20,C5,7E,E5,21,20,00,48,06
810 DATA 1DD8
820 DATA 00,ED,42,CB,25,CB,25,11
,48,00,19,CB,27,CB,27,2F,CE,BA,4
4,4D,5F,16,00,CD,2A,DD,E1,23,C1,
10,D9,CD,9F,00,67,3E,20,BC,20,23
,3A,62,C0,1E,3D,BB,20,0A,3E,BA,1
E,01,32,62,C0,C3,14,DD,3A,E7,C0,
1E,01,BB,C2,F1,D3,3E,02,32,E7,C0
,C3,B6,DA,3E,1D,BC,20,1D,3A,61
830 DATA 222B
840 DATA C0,47,3E,47,B8,20,08,3E
,C3,32,61,C0,C3,14,DD,3A,61,C0,0
E,04,91,32,61,C0,C3,14,DD,3E,1C,
BC,20,1C,3A,61,C0,47,3E,C3,B8,20
,08,3E,47,32,61,C0,C3,14,DD,3A,6
1,C0,C6,04,32,61,C0,C3,14,DD,3E,
1E,BC,20,19,3E,F9,32,EE,C0,18,0B
,3E,1F,BC,C2,F9,DB,3E,01,32,EE
850 DATA 228D
860 DATA C0,21,67,C0,3A,E7,C0,1E
,01,BB,28,03,21,A7,C0,3A,62,C0,1
E,3D,BB,28,04,11,20,00,19,3A,61,
C0,CB,3F,CB,3F,D6,11,4F,06,00,09
,E5,3A,EE,C0,1E,F9,BB,28,0A,7E,1
E,00,BB,20,19,E1,C3,F9,DB,16,1E,
3A,62,C0,1E,3D,BB,20,02,16,0F,7E
,5A,BB,20,04,E1,C3,F9,DB,7E,CB
870 DATA 2150
880 DATA 27,CB,27,2F,3C,3C,F5,3A
,62,C0,4F,F1,89,3D,3D,5F,16,00,3
A,61,C0,3C,4F,06,00,CD,58,DD,E1,
7E,F5,3A,EE,C0,1E,F9,BB,20,04,F1
,3C,18,02,F1,3D,77,2A,E9,C0,26,0
0,ED,4B,EE,C0,09,54,5D,ED,4B,E8,
C0,CD,2A,DD,C3,F9,DB,21,1C,1B,3A
,62,C0,C6,02,CD,4D,00,23,3A,61
890 DATA 250D
900 DATA C0,3D,CD,4D,00,C3,F9,DB
,ED,43,E8,C0,ED,53,E9,C0,06,04,C
5,ED,4B,E8,C0,06,00,CD,11,01,CD,
20,01,CD,FC,00,CD,20,01,CD,FC,00
,CD,20,01,CD,FC,00,CD,20,01,C1,1
3,10,DD,C9,ED,43,E8,C0,ED,53,E9,
C0,06,04,C5,ED,53,E9,C0,ED,4B,E8
,C0,06,00,CD,11,01,CD,4A,00,2F
910 DATA 2924
920 DATA CD,4D,00,ED,4B,E8,C0,ED
,5B,E9,C0,06,00,16,00,CD,11,01,C
D,20,01,CD,FC,00,CD,20,01,CD,FC,
00,CD,20,01,CD,FC,00,CD,20,01,CD
,11,01,CD,4A,00,CD,11,01,CD,4A,0
0,2F,CD,4D,00,C1,13,10,B3,C9,3E,
07,1E,EE,CD,93,00,3E,01,32,F0,C0
,32,F3,C0,32,F2,C0,3E,00,32,F1
930 DATA 228A
940 DATA C0,3A,FF,BF,2E,10,BD,C2
,D5,DD,3E,04,C3,E2,DD,2E,0C,BD,C
2,E0,DD,3E,03,C3,E2,DD,3E,02,4F,
3E,09,1E,10,CD,93,00,3E,08,1E,00
,CD,93,00,3E,0C,1E,00,CD,93,00,3
E,0B,1E,1E,CD,93,00,3E,08,1E,00,
CD,93,00,3E,00,F5,3E,0D,1E,00,CD
,93,00,41,3E,00,32,9E,FC,3A,66
950 DATA 1EA2
960 DATA C0,CB,3F,CB,3F,6F,3A,9E
,FC,BD,DA,1E,DE,10,EA,F1,3C,F5,1
E,03,BB,C2,09,DE,CD,B7,00,DA,5A,
D6,21,FF,BF,3A,F0,C0,1E,01,BB,28
,03,21,00,C0,7E,5F,3A,F1,C0,BB,2
8,06,3C,32,F1,C0,18,52,3E,01,32,
F1,C0,3A,F0,C0,1E,01,BB,20,1C,3A
,64,C0,5F,3A,F3,C0,BB,28,06,3C
970 DATA 26B3
980 DATA 32,F3,C0,18,35,3E,02,32
,F0,C0,3E,00,32,F3,C0,18,DC,3A,6
5,C0,5F,3A,F3,C0,BB,28,06,3C,32,
F3,C0,18,19,3E,01,32,F0,C0,3E,01
,32,F3,C0,3A,EB,C0,5F,3A,F2,C0,B
B,CA,5A,D6,3C,32,F2,C0,3E,00,32,
9E,FC,21,10,C0,3A,F0,C0,1E,01,BB
,28,03,21,40,C0,3A,F1,C0,5F,16
990 DATA 2506
1000 DATA00,19,7E,1E,C8,BB,28,1C
,3E,01,1E,0F,CD,93,00,3E,08,1E,1
0,CD,93,00,3E,0C,1E,04,CD,93,00,
3E,0D.1E,00,CD,93,00.7D,D6.10,6F
,16,00,7E,1E,C8,BB,CA,EF,DE,16,0
1,7A,32,F4,C0,7D,C6,20,6F,16,00,
7E,1E,C8,BB,CA,02,DF,16,01,7A,32
,F5,C0,3E,00,32,FA,C0,3A,FA,C0
1010 DATA 1F0E
1020 DATA3C,32,FA,C0,3E,00,32,F6
,C0,32,F7,C0,3A,F4,C0,1E,01,BB,2
0,14,21,66,C0,3A,FA,C0,85,6F,7E,
32,F6,C0,7D,C6,20,6F,7E,32,F7,C0
,3A,F5,C0,1E,01,BB,20,1D,21,A6,C
0,3A,FA,C0,85,6F,7E,5F,3A,F6,C0,
BB,D2,5B,DF,7B,32,F6,C0,7D,C6,20
,6F,7E,32,F7,C0,3A,F7,C0,5F,3E
1030 DATA 2A34
1040 DATA1E,93,32,F7,C0,3A,F7,C0
,5F,3E,06,CD,93,00,3A,F6,C0,5F,3
E,09,CD,93,00,3A,EC,C0,16,00,5F,
06,03,CB,23,CB,12,10,FA,21,00,00
,3A,66,C0,CB,3F,CB,3F,47,3A,9E,F
C,B8,D2,30,DE,23,CD,20,00,28,02,
18,F1,3A,FA,C0,4F,3E,20,B9,C2,0B
,DF,3A,66,C0,CB,3F,CB,3F,47,3A
1050 DATA 2343
1060 DATA9E,FC,B8,D2,30,DE,18,F7
,21,F9,BF,06,06,36,20,23,10,FB,2
6,01,2E,01,CD,C6,00,21,91,E0,CD,
0A,DA,21,F9,BF,06,06,3E,02,32,AF
,FC,CD,9F,00,F5,3E,01,32,AF,FC,F
1,1E,0D,BB,CA,0B,E0,1E,20,BB,DA,
D6,DF,77,DF,23,10,E0,3E,02,32,AF
,FC,CD,9F,00,F5,3E,01,32,AF,FC
1070 DATA 266C
1080 DATAF1,1E,0D,BB,C2,F6,DF,3E
,00,CD,EA,00,21,F9,BF,06,06,C5,E
5,7E,CD,ED,00,E1,C1,23,10,F5,CD,
F0,00,3E,00,CD,EA,00,21,FF,BF,06
,EE,C5,E5,7E,CD,ED,00,E1,C1,23,1
0,F5,CD,F0,00,C3,F1,D3,3E,07,1E,
B8,CD,93,00,CD,E1,00,21,F9,BF,06
,06,C5,E5,CD,E4,00,DA,7B,D2,E1
1090 DATA 2CDA
1100 DATAC1,77,23,10,F2,CD,E7,00
,26,01,2E,01,CD,C6,00,21,91,E0,C
D,0A,DA,06,06,21,F9,BF,7E,DF,23,
10,FB,CD,E1,00,21,FF,BF,06,EE,C5
,E5,CD,E4,00,DA,7B,D2,E1,C1,77,2
3,10,F2,CD,E7,00,C3,F1,D3,4E,4F,
4D,42,52,45,3A,00,00,00,00,00,00
,00,00,00,00,00,00,00,00,00,00
1110 DATA 2126
```

CONCURSO MENSUAL

PREMIO

MSX-TEST



Un soft a elección entre IDEA BASE, IDEA TEXT, o BASIC TUTOR.

CONCURSO MENSUAL

Auspiciado por TELEMATICA S.A. fabricante en Argentina de las computadoras personales TALENT MSX

Para participar en este certamen deben señalar cual es la información correcta que presenta cada ítem. Junto con las respuestas deben remitir los datos en el correspondiente cupón. Los premios se entregarán en la administración de la editorial. Los que no puedan concurrir pueden solicitar el envio pagando el franqueo contrareembolso. Los premios podrán ser reclamados dentro de los 120 días después de haber sido anunciados.

Ganador del MSX-TEST No.11

Carlos J. Caccia

Respuestas correctas

1)a. Memoria RAM de video. 2)b. Tipografía para sacar archivos por impresora. 3)b De pantalla. 4) b.Un trazador de gráficos. 5)b. Video Display Processor

MSX-TEST No 15-CIERRE 1 DE MAYO DE 1989

1)¿Qué notación musical se utiliza en la MSX?
a. alfabética
b. alfanumérica
c. notación de Weisner

2)¿Cuántos utilitarios nuevos incorpora el MSX-DOSTOOLS?
a. 20
b. 28
c. 36

3)¿Cuál es el archivo que, en caso de ser hallado, carga y ejecuta automáticamente un programa?
a. AUTOEXEC.BAS
b. COMMAND.COM
c. MSXDOS.SYS

4)¿Qué tipo de joystick es el WARP I?
a. mecánico
b. electrónico
c. electromecánico

5)¿Cuántos canales de sonido se pueden utilizar en la MSX?
a. tres
b. cinco
c. dieciséis

Nombre y apellido: ____________________

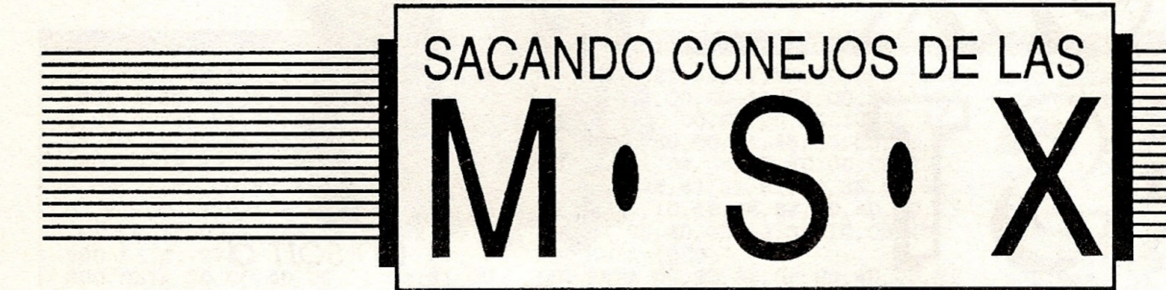
Dirección: ____________________

Documento: __________ Edad: ______ Máquina: __________

Qué es lo que más le gusta de la revista: ____________________

Qué le agregaría: ____________________

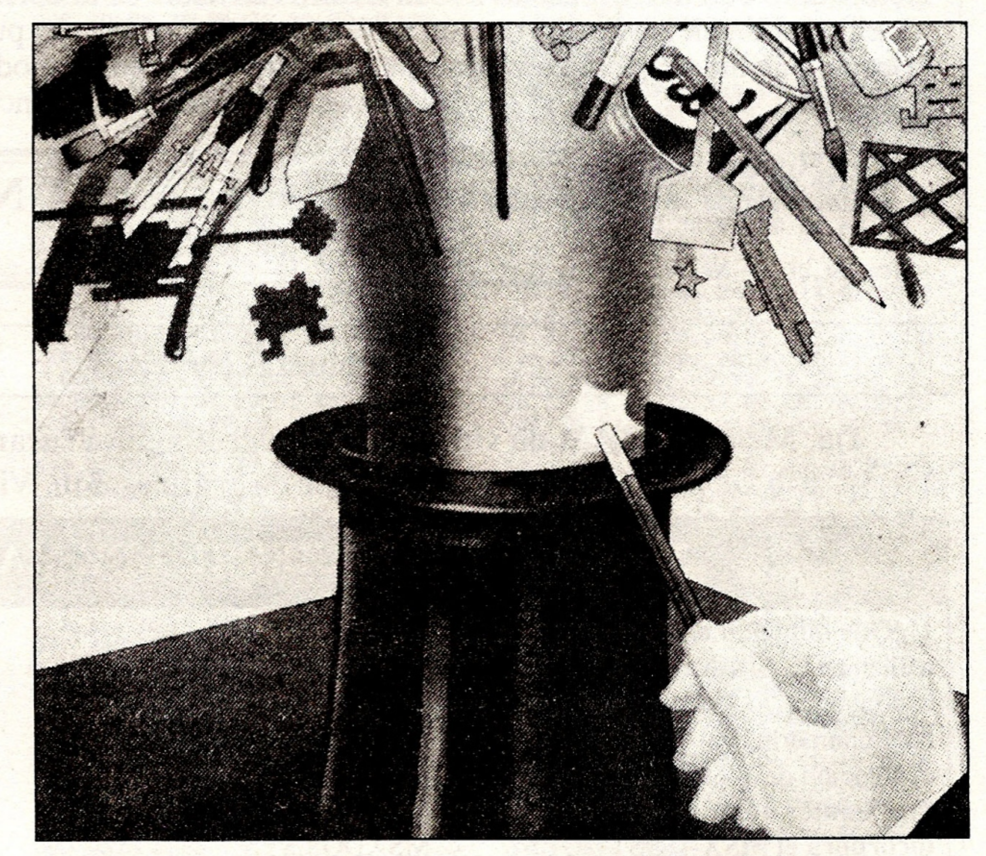
Que es lo que no me gusta: ____________________



SACANDO CONEJOS DE LAS MSX

Como si fuera la galera de un mago, nuestra computadora brinda la oportunidad de sacar un truco tras otro. Desde provocar y anular sonidos hasta escribir en"italic" pasando por la decoración de las carátulas de pantallas.

Ofrecemos en estas páginas una colección de trucos, rutinas y pequeños programas para las computadoras MSX. Los mismos nos permitirán conocer muchas posibilidades que nos brinda nuestra computadora.



1)Pantalla por páginas
Con este truco podremos ver la pantalla en el modo texto, página por página, exceptuando las páginas 2 y 3.

```
10 INPUT "ENTRA UN VALOR DEL 0 AL 15, PERO QUE SEA DISTINTO A 2 y 3";X
20 BASE(0)=X*1024 : VDP(2)=X : POKE&HF23,X*4
```

2)Rapidez en alta resolución
Veamos las diferencias entre el tiempo de ejecución de estos dos listados. La clave de la rapidez está en la instruccion VPOKE

LISTADO 1

```
10 SCREEN 2
20 FOR J=1 TO 192
30 FOR I=1 TO 255 STEP 8
40 LINE(I,J)-STEP(8,0),X
50 X=X+1 : IF X>15 THEN X=1
60 NEXT I,J
70 X $=INPUT$(1)
```

LISTADO 2

```
10 SCREEN 2
20 COLOR 1,1
30 X=0
40 FOR I=8192 TO 14335 STEP 8
50 FOR J=0 TO 7
60 VPOKE I+J,X+J
70 NEXT J
80 X=X+1
90 IF X+248 THEN X=0
100 NEXT I
110 X$=INPUT4$(1)
```

3)Visualización de las teclas de función
Para ver las sentencias definidas para las teclas de función debemos colocar POKE&HF3DE,1. Para que desaparezcan, el truco es POKE&HF3DE,0. La aparición y desaparición se produce recién después de colocar CLS o presionar la tecla SHIFT.

4)Cuestión de ruidos
Con POKE&HF3DB, 1 escuchare-

mos un eco luego de apretar cada tecla. Con POKE&HF3DB, 0 ese ruido desaparece.

5)Secuencias de escape

Estas funciones están disponibles desde el PRINT del BASIC, la rutina de BIOS CHPUT, CONOUT del BIOS del MSX-DOS llamado directamente y CONSOLE OUTPUT de la función del MSX-DOS.

[ESC] A Cursor arriba
[ESC] B Cursor abajo
[ESC] C Cursor derecha
[ESC] D Cursor izquierda
[ESC] H Cursor a home
[ESC]Y [Fila+20H] [Columna+20H] función locate
[ESC] E Borrar pantalla
[ESC] K Borrar hasta el fin de la línea
[ESC] J Borrar hasta el fin de la pantalla
[ESC] U Borrar toda la línea
[ESC] L Insertar una línea
[ESC] M Eliminar una línea
[ESC]x4 Coloca el cursor relleno
[ESC]x5 Apaga cursor
[ESC]y4 Coloca el cursor pequeno
[ESC]y5 Prende cursor

6)Locate

Si queremos utilizar el MBASIC y su respectivo compilador, nos encontramos que no tenemos la sentencia LOCATE que permite ubicar el cursor en la pantalla. Para poder simular esta sentencia podemos definir la siguiente función:

```
10DEFFNCURS(X,Y)=CHR$(27)
+"Y"+CHR$(X+32)+CHR$(Y+32)
```

y luego utilizarla con la sentencia PRINT:

```
18PRINT FNCURS(10,10);"ESTO
ES UN TEST"
```

7)Input diferente

Esta rutina anula la aparición del signo ? en el INPUT. Para recuperar el INPUT normal debemos colocar POKE &HFDE0,&HC9

```
10 POKE &HFDE1,&HC3
20 POKE&HFDE2,&HD2
30 POKE &HFDE3,&H23
40 POKE &HFDE0,&HF1
```

8)Código de caracteres

Con esta rutina podemos ver el código decimal de cada tecla que pulsemos.

```
10 CLS:KEY OFF:SCREEN 0:CO-
LOR 11,1,1
20 A$=INKEY$:IF A$="" THEN20
30 FOR I=0 TO 255
40 IF A$=CHR$(I) THEN PRINT
"El CODIGO DECIMAL DE
";CHR$(I): PRINT "ES: ";I
50 NEXT
60 GOTO 20
```

9)Interrupciones

Si nos encontramos con algún programa largo, para ahorrar tiempo cortemos las interrupciones a fin de leer el teclado y, en consecuencia, la actualización de la variable TIME.

Estas funciones, si bien son realizadas con rapidez, también devoran nuestro tiempo. Con VDP(1)=208 se desactivan las lecturas del teclado. Aquí el sistema no recibe ninguna instrucción desde el teclado, por lo tanto las teclas STOP o CONTROL/STOP no provocarán ningún efecto.

Para volver a activar las lecturas del teclado, hace falta introducir VDP(1)=204

10)Gráfico I:

Este gráfico nos puede resultar útil para decorar carátulas de pantallas.

```
10AS="s3m7800c4r8o3q18al8b18c
4r8o3q18a18b18c4r2r8o5c8c4.b8d4.c
8q2f4"
20 PLAY AS
30 GOTO 30
```

11)Gráfico II:

Para mejorar la presentación de los programas podemos utilizar el siguiente listado.

```
10 SCREEN2: DEFINT A-Z: CO
LOR 1,15,11
20 FOR A=1 TO 255 STEP 19: FOR
B=1 TO 191 STEP 11
30 FOR C=A/2 TO B STEP 7: FOR
D=B*2 TO 1 STEP 38
40 LINE STEP (A,C)-(D,B),13
50 LINE STEP (D,C)-(B,A),12
60 LINE STEP (C,A)-(B,D),11
70 LINE STEP (A,C)-(B,D),10
80 NEXT
90 NEXT
100 NEXT
110 NEXT
120 Z$=INPUT$(1)
```

12)Paleta

BASE es el nombre de una instrucción poco usada. Se puede utilizar, entre otras funciones, para crear nuestra propia configuración de la memoria RAM de visualización. Por ejemplo, la paleta de colores comienza en la dirección 8192 de la VRAM y tiene una longitud de 32 caracteres. Este programita nos permitirá ver la pantalla con varios caracteres de diferentes colores.

```
10 COLOR 1,15
20 SCREEN1
30 FOR N=8192 TO 8192*32
40 VPOKE N,INT(RND(1)*255)
50 NEXT N
```

13)Caracteres

Esta rutina nos permite modificar la definición de los caracteres. De esta manera, podremos tener siempre cargados en nuestra computadora los más diversos diseños de letras y números.

```
10 SCREEN0
20 PRINT "Esto es una prueba"
30 FOR T=1 TO 1300: NEXT
40 FOR F=0 TO 7
50 READN
60 VPOKE(BASE(2)+(97*8+F)),N
70 NEXT
80 PRINT "Se cambio la definicion
del caracter -a-"
90 DATA 0,255,255,0,0,31,31,0
```

14)16K más

Esta rutina le dá uso a 16K de memoria de la página 1 del slot 1, que no se emplea y permanece oculta cuando se trabaja con BASIC.

Una vez ejecutada, quedan definidas dos teclas: F3 y F8.

Una función nos permite grabar el contenido de la memoria de video en modo texto; la otra lee dicho contenido y lo presenta en pantalla. Al tipear F8 el cursor espera que ingresemos el

número de pantalla que se ha de escribir o leer. En total se pueden almacenar 16 pantallas (de 0 a 15).
La rutina se puede usar con el drive conectado, ya que no produce ninguna interferencia. Además, permite trabajar con dos o más programas en BASIC simultáneamente, sin necesidad de acceder al disco o casete.

```
10 WIDTH 38
20 DEF USR8=51770!
30 DEF USR9=51810!
40 KEY3,"LEC=USR9(0)"+CHR$(29)+CHR$(29)
50 KEY 8,"ESC=USR8(0)"+CHR$(29)+CHR$(29)
110 FOR R=51750! TO 51839!
120 READ D: POKE R,D: NEXT R
130 NEW
140 DATA64,68,72,76,80,84,88,92,96,100,104,108,112,116,120,124,0,0,0,0
150 DATA1,12,0,33,0,0,62,32,205,86,0,33,38,202,58,248,247,133,111,30,0,86,33,0,0,1,192
160 DATA3,62,84,211,168,205,89,0,62,80,211,168,201,33,38,202,58,248,247,133,111,126,103,46,0,17,0,0,1,192,3,62,84,211,168,205,92,0,62,80,211,168,201
```

15)Multiplicación
Con este programita podremos ver las tablas de multiplicación

```
10 CLS
20 FOR K=10 TO 20
30 CLS
40 FOR I=1 TO 20
50 J=K*I
60 PRINT K;" x ";I;" = ";J
70 FOR D=1 TO 100
80 NEXT D
90 NEXT I
100 FOR A=1 TO 400
110 NEXT A
120 NEXT K
```

16)Sprite sin sprite
Con el siguiente listado mostramos otro camino para que los programadores puedan manejar sprites. En la línea 50, el valor 195 define la forma del carácter que se va a mover luego como un sprite. Este valor puede ser modificado para cambiar su forma. En tanto la variable F define cuál es el byte que será definido. Para definir todo un carácter es necesario utilizar 8 bytes, de ahí el bucle "FOR F=A TO A+7"
El sprite definido está en las 8 primeras posiciones de la tabla de diseño de sprites (BASE 9).
Para correr el sprite modifiquemos las dos primeras posiciones de la tabla de atributos de sprites (BASE 8). Con la primera posición se consigue variar la coordenada "X" del sprite, mientras que la segunda varía la coordenada "Y".

```
10 SCREEN 1
20 KEY OFF
30 A=BASE(9)
40 FOR F=A TO A+7
50 VPOKE F,195
60 NEXT
70 S=BASE(8)
80 FOR I=0 TO 200
90 VPOKE S,I
100 VPOKE S+1,I
110 NEXT
120 GOTO 80
```

17)Redefiniendo teclas
Al comenzar a manejar nuestra MSX, las teclas de función se encuentran definidas con funciones como RUN, COLOR, LIST, etcétera . Con este listadito redefiniremos las funciones que les asignamos a dichas teclas. Las nuevas funciones serán:
F1=moverá el curso siguiente.
F3=moverá el cursor al último carácter de la línea actual.
F4=borra la línea actual.
F5=borra el carácter siguiente al cursor.
F6=genera un "bip".
F7=borra el carácter anterior al cursor.
F8=mueve el cursor a la próxima tabulación horizontal, borrando por donde pasó.
F9=borra la pantalla y el cursor se coloca en el ángulo superior izquierdo.
F10=borra el carácter donde está el cursor.
Estas definiciones son muy útiles cuando nos encontramos editando programas.

```
10 CLS
20 FOR F=1 TO 10
30 READ A
40 KEY F,CHR$(A)
50 NEXT
60 DATA 2,6,14,21,5,7,8,9,12,127
```

18)Caracteres invertidos
Este programa convierte todos los carcteres en efecto invertido. Funciona únicamente mientras no pasemos del modo de pantalla 1 a cualquier otro:

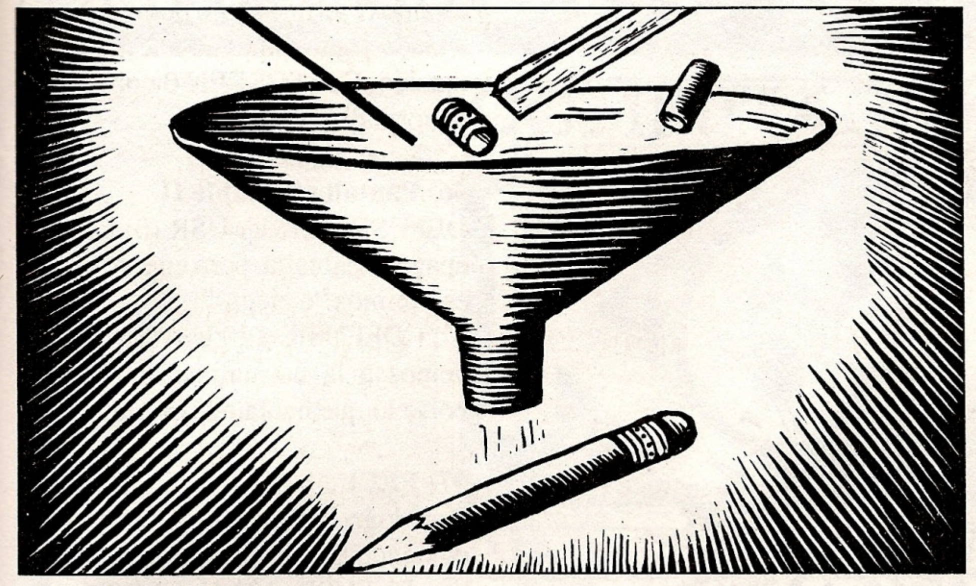
```
10 SCREEN 1
20 CLS
30 FOR F=0 TO 2048
40 IF F=256 THEN F=264
50 D$="00000000"
60A$=RIGHT$(D$+BIN$(VPEEK(F)),8)
70 FOR B=110;NEXT B
120G$=V$(1)+V$(2)+V$(3)+V$(4)+V$(5)+V$(6)+V$(7)+V$(8)
130 VPOKE F,VAL(" &b"+G$)
140 NEXT F
```

19)Rótulos
Con esta pequeña rutina podemos darnos cuenta de la utilidad de la sentencia MID$. Hace que un rótulo se desplace de derecha a izquiertrucciones de los programas.

```
10 CLS
20 KEY OFF
30 SCREEN 0: WIDTH 40
40 A$="ESTO LO PUEDES INCORPORAR A TUS PROGRAMAS - LOAD MSX"
60 LOCATE 0,10
70 PRINT MID$(A$,B,40)
80 FOR F=1 TO 30: NEXT
90 B=B+1
100 IF B=43 THEN B=1
110 GOTO 60
```

20) Efecto
Vemos con este listado un bonito efecto de pantalla, con líneas de diferentes colores. Los puntos aleatorios de las rayas se definen en las líneas 40,50,60 y 70

```
10 SCREEN 2
```



20 A=1
30 FOR T=1 TO 50000!
40 X=INT(RND(1)*255)
50 Z=INT(RND(1)*255)
60 C=INT(RND(1)*190)
70 S=INT(RND(1)*190)
80 LINE (X,C)-(Z,S),A
90 A=A+1
100 IF A=16 THEN A=1
110 NEXT T

21)Enfilar y encolumnar
Esta rutina permite numerar en los márgenes izquierdo e inferior de la pantalla del modo texto 0 las filas y columnas en que puede dividirse la pantalla normalmente

```
10 SCREEN 0 : WIDTH 37
20 CLS : KEY ON
30 FOR F=0 TO 23
40 VPOKEF*40,VAL(MID$(STR$
(F),LEN(STR$(F)),1))+48
50 NEXT F
60 FOR C=0 TO 39
70VPOKE920+C,VAL(MID$(STR$
(C),LEN(STR$(C)),1))+48
80 NEXT C
```

22)Protección:
Para proteger los programas de los piratas tengamos en cuenta los siguientes POKES:
POKE 65297,195
POKE 65298,0
POKE 65299,0

23)Nombres I
Con este listado podremos averiguar el nombre del último programa que hemos cargado

```
10 FOR I=63600! TO 63607!
20 N=PEEK(I)
30 PRINT CHR$(N)
40 NEXT I
```

24)Nombres II:

Si queremos saber el nombre del último programa que hemos grabado, probemos con el siguiente listado.

```
10 FOR I=63590! TO 63596!
20 N=PEEK (I)
30 PRINT CHR$ (N)
40 NEXT 1
```

25) Ventana
Es sabido que la pantalla tiene 23 líneas. Pero esta cantidad podemos cambiarla fácilmente con la sentencia: POKE 62385,N donde N es la cantidad de líneas que tendrá de largo la pantalla.

26) CLS
Para borrar la pantalla (CLS) de una manera diferente podemos ingresar la siguiente sentencia.
DEFUSR=&HC3
A=USR (0)

27) Efecto luz
Con esta rutina podemos cambiar el estado de la luz de la tecla CAPS. OUT 170,103 apaga la luz y OUT 170,7 la enciende. Pero esto no cambia el modo de escritura, sólo el estado del led indicador.

```
10 OUT 107,103
20 A=A+9: FOR I=1 TO A: NEXT 1:
BEEP
30 OUT 170,7
40 E=E+20: FOR I=1 TO E: NEXT I
```

28) OFF-ON
Para eliminar o escribir la última línea de la pantalla con las teclas de función, el BASIC cuenta con las sentencias KEY OFF, KEY ON, respectivamente.
Otra forma es con:
DEFUSR=&HCA : A=USR (0), que elimina la línea y
DEFUSR=&HCD : A=USR1(0) que la vuelve a escribir.

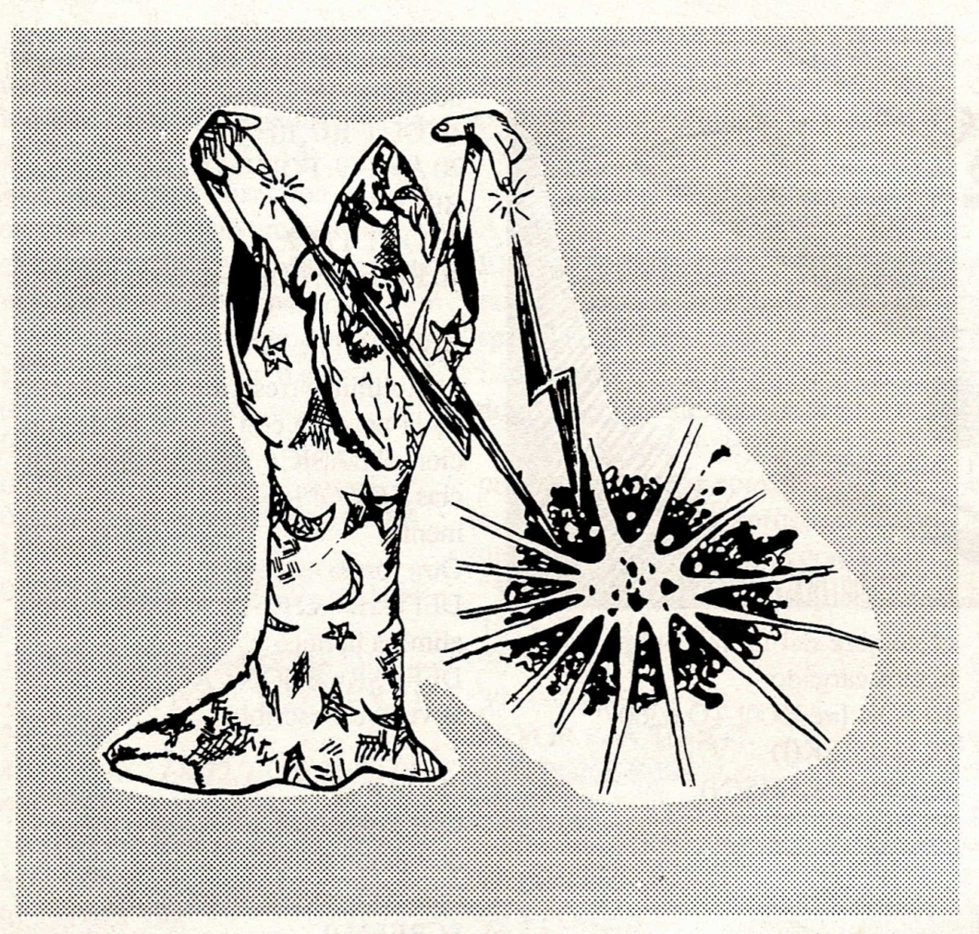
29) FONT
Con esta rutina se puede cambiar la definición de los caracteres.
Para recuperarla, ingresemos:
SCREEN 0

```
10 FOR N=BASE (2)+8*33 TO
(BASE(2)+8*64)+255
20 PRINT "Revista K-64"
30 VPOKE N,248 : NEXT N
```

30) Bloqueo
Para provocar un bloqueo intencional de la computadora utilicemos la siguiente instrucción:
POKE 64922, 195.
Para recuperar el control de la máquina deberemos apagarla.

31) Italic
Esta rutina permite escribir en pantalla con tipografía "italic", al tiempo que mantiene la clásica tipografía de la computadora.
Asegurémonos que la computadora está en modo minúsculas.
Para escribir con letras "italic minúsculas" se debe presionar simultáneamente GRAPH y la tecla deseada.
Para lograr letras "italic mayúsculas" se debe presionar simultáneamente CODE y la letra deseada.

```
10 SCREEN 1
20 FOR F=0 TO 93
30 Q=8 : W=0 : E=1 : READ M : FOR
```



C=0 TO 7: VPOKE
M*8+C,VPEEK((F+33)*8+C) /0
40 IF W=1 THEN Q+Q/2
50 SWAP W,E
60 NEXT
70 NEXT
80 END
90 DATA159,185,191,155,152,225,
5,135,235,231,233,243,23,242,29,9,172,
171,186,239,189,244,251,236,7,183,6,
134,241,166,167
100DATA217,132,151,141,139,140,
148,129,177,161,145,179,181,230,164,162,
163,131,147,137,150,10,149,136,138,160,133,
1,30,13,224,238,187,196,17,188,199,205,20,
21,19,220,198,221,200,11,27
110 DATA194,219,204,24,210,18,
192,26,207,28,25,15,237,173,218,229

32) Buscadora
La sentencia
10 DEFUSR=&HDF : A=USR(0) :
MOTOR OFF
busca programas grabados en cinta.

33) Mayúsculas I
Con esta rutina los programas se independizan del tipo de letras que ingrese el usuario, pues ya sean mayúsculas o minúsculas, indistintamente, son controladas en forma automática, garantizando el ingreso en mayúsculas.

```
10 PRINT "Mensaje pidiendo ingreso por teclado"
20 LINE INPUT A$
30 FOR L=1 TO LEN (A$)
40 LET COD= ASC (MID$(A$,L,1))
50 IF COD>=97 AND COD<=122 THEN LET COD=COD-32
60LETAUX$=AUX$+CHR$(COD)
70 NEXT L
80 LET A$=AUX$
90 PRINT A$
100 REM RETURN
```

34) Mayúsculas II
En aquellos casos en que se necesite que tanto las entradas de datos por teclado como la salida por pantalla sea en mayúsculas podemos emplear el siguiente poke.
POKE 64683,255 fija las mayúsculas.
POKE 64683,0 retorna al modo normal.

35) Pantalla invisible I
DEFUSR=63:U=USR(0) desactiva la pantalla y no podemos ver lo que estamos escribiendo, a pesar de que el teclado sigue tomando nuestra información. Con SCREEN 0 volvemos a la normalidad.

36) Pantalla invisible II
DEFUSR=121: U=USR (0) también apaga la pantalla, pero en este caso sí escribimos "a ciegas".
Con DEFUSR=120:U=USR(0) volvemos a la normalidad sin que se borre lo que habíamos escrito.

37) BEEP
Para lograr un efecto similar al del comando BEEP tendremos que colocar: DEFUSR=192:U=USR(0)

38)Modos de pantalla
Con DEFUSR=106:U=USR(0) pasamos al modo de pantalla 1.
Con DEFUSR=108:U=USR(0) volvemos al modo de pantalla 0.

39) Nada se pierde
Supongamos que tenemos un programa en la memoria y luego lo borramos, ya sea con el clásico NEW o con la rutina de RESET (DEFUSR=0: PRINT USR(0)).
Ahora supongamos que luego de "borrarlo" queremos recuperarlo. Lo conseguiremos ingresando las siguientes sentencias en este orden:

```
POKE 32768,0
POKE 32769,255
POKE 32770,255
LIST
1 REM
SAVE"CAS:NAME"
```

40) Direcciones

En las direcciones 64586 y 64587 se almacena el valor del comienzo del área del sistema. Esta se calcula con PRINTPEEK(64586)+256*PEEK(64587).
En las direcciones 64584 y 64585 se almacena el valor del comienzo de la RAM que usa el BASIC. Se calcula por el mismo procedimiento que la anterior. En las direcciones 64670 y 64671 se guarda el valor de la variable TIME. Se calcula de la misma forma.

SIMULTANEO CON EE.UU.
SUPLEMENTO BYTE EN K64



La revista líder de computación de la Argentina se asocia con

BYTE

La revista líder mundial en computación

AHORA EN CASTELLANO



Ahora los lectores de K64 podrán acceder a los mejores artículos sobre software, hardware, notas, novedades sorprendentes y todo lo último sobre PC publicado por BYTE (U.S.A.) y, como siempre, notas trucos y programas para usuarios de la norma MSX.



K64 COMPUTACION PARA TODOS - Editorial Proedi S.A., Paraná 720, 5º piso, (1017) Buenos Aires, Tel. 46-2886/49-7130

EXPLORADOR DE CABEZALES



Autor: Patricio G. Expósito
Tipo: Utilitario
Computadora: MSX

Desde el surgimiento de la norma MSX ya ha pasado bastante tiempo y cada día son más los usuarios de estas computadoras que poseen una unidad de discos. Cuando se encaran trabajos de cierta seriedad hay dos ventajas por las que conviene pagar la diferencia entre la disquetera y la unidad de casete: velocidad y seguridad.

Sin embargo, no son pocos los usuarios que siguen usando su computadora en forma "casera" y, para muchos de ellos, no se justifica pagar el precio de la disquetera.

Un problema que se les suele presentar es que muchos utilitarios están hechos exclusivamente para ser usados con disquetera, o bien no pueden ser aprovechados al cien por ciento de su capacidad si vienen en casete.

El utilitario que presentamos aquí rompe la regla, ya que está hecho para los "sufridos" usuarios de casetes.

El programa básicamente puede hacer cuatro cosas: explorar, dar información de la unidad de cinta, leer un archivo y grabar un archivo.

La exploración, a su vez, se divide en lectura y búsqueda.

La información de la unidad de cinta puede ser dada por pantalla o impresora, a requerimiento del usuario. Consta de listados de programas, información del programa, búsqueda del mismo más una opción adicional de borrar datos.

Este programa, por su extensión y utilidad, seguramente será muy bienvenido por la numerosa grey de MSX.

```
10 KEY 1,"DEFUSR=0:A=USR("
20 '
30 'EXPLORADOR DE CINTAS
40 '
50 CLEAR 1000,&HD9FF
60 ON STOP GOSUB 1400
70 SCREENO:WIDTH 40:PRINT"    EXF
LORADOR DE CABECERAS PATOSOFT"
80 DIM TP$(4),PR$(100),TP(100),I
N(100),FI(100),EJ(100)
90 TP$(1)="BLOAD":TP$(2)="LOAD":
TP$(3)="CLOAD"
100 FOR T=&HDA00 TO &HDA44:READ
A$:POKE T,VAL("&H"+A$):S=S+VAL("
&H"+A$):NEXT:DEFUSR=&HDA00
110 IF S-8966 THEN PRINT "ERROR
EN DATAS":STOP
120 KEY OFF
130 'MENU
140 GOSUB 970
150 PRINT "MENU DE OPCIONES"
155 PRINT "*            ---- -- -
--------"
160 X=30:Y=10:GOSUB 1060
170 LOCATE 5,10:PRINT " 1 EXPLOR
AR"
180 LOCATE 5,12:PRINT " 2 INFORM
ACION U. C."
190 LOCATE 5,14:PRINT " 3 LEER A
RCHIVO"
200 LOCATE 5,16:PRINT " 4 SALVAR
 ARCHIVO"
210 LOCATE 5,18:PRINT " 5 FIN"
220 NO=5:GOSUB 1210
230 ON OP GOSUB 270,560,1440,147
0
240 IF OP-5 THEN 130
250 KEY ON:CLS:END
260 '
270 'EXPLORA
280 '
290 GOSUB 970:PRINT "EXPLORACION
":LOCATE ,8
300 X=30:Y=6:GOSUB 1060
310 LOCATE 6,10:PRINT "1 LEE"
320 LOCATE 6,12:PRINT "2 BUSCA"
330 LOCATE 6,14:PRINT "3 SALIDA"
340 NO=3:GOSUB 1210
350 IF OP=3 THEN OP=0:RETURN
360 N$="":IF OP=2 THEN GOSUB 131
0
370 CLS:PRINT"PLAY":PRINT ""
380 IF NP=100 THEN BEEP:PRINT "M
EMORIA LLENA":MOTOR OFF:GOSUB 12
70:RETURN
390 A$=USR(STRING$(20,0))
400 IF ASC(A$)=0 THEN 390
410 T=ASC(A$)
420 IF T=&HD THEN MOTOR OFF:RETU
RN
430 NP=NP+1:TP(NP)=0
440 IF T=&HD0 THEN TP(NP)=1
450 IF T=&HEA THEN TP(NP)=2
460 IF T=&HD3 THEN TP(NP)=3
470 PR$(NP)=MID$(A$,2,6)
480 IN(NP)=ASC(MID$(A$,8,1))+ASC
(MID$(A$,9,1))*256
490 FI(NP)=ASC(MID$(A$,10,1))+AS
C(MID$(A$.11.1))*256
```

```
500 EJ(NP)=ASC(MID$(A$,12,1))+AS
C(MID$(A$,13,1))*256
510 PRINT "FOUND: "PR$(NP)"    "T
P$(TP(NP))
520 IF NP=100 GOTO 380
530 IF N$<>PR$(NP) GOTO 390
540 PRINT "HALLADO":MOTOR OFF:GO
SUB 1270
550 RETURN
560 'INFORMACION ULTIMA CINTA
570 GOSUB 970:PRINT "INFORMACION
 U.C."
580 X=39:Y=10:GOSUB 1060
590 LOCATE 1,10:PRINT " 1 LISTA
PROGRAMAS"
600 LOCATE 1,12:PRINT " 2 INFORM
ACION PROGRAMA"
610 LOCATE 1,14:PRINT " 3 BUSCA
PROGRAMA"
620 LOCATE 1,16:PRINT " 4 BORRAR
 DATOS"
630 LOCATE 1,18:PRINT " 5 SALIDA
"
640 NO=5:GOSUB 1210
650 ON OP GOSUB 680,820,1110,118
0
660 IF OP<5 GOTO 560
670 OP=0:RETURN
680 CLS:IF NP=0 THEN BEEP:PRINT
"NO HAY PROGRAMAS EN MEMORIA":GO
TO 800
690 PRINT"(I)mpresora o (P)antal
la?"
700 J$=INKEY$
710 IFJ$<>"I"ANDJ$<>"i"ANDJ$<>"p
"ANDJ$<>"P"THEN700
720 IFJ$="i"ORJ$="I"THENGOSUB150
0
730 CLS
739 PRINT "PROG. NOMBRE  CARGA":
PRINT
740 FOR T=1 TO NP
760 LOCATE0,T+2:PRINTT
770 LOCATE6,T+2:PRINTPR$(T)
780 LOCATE14,T+2:PRINTTP$(TP(T))
790 NEXT T
800 GOSUB 1270
810 RETURN
820 LOCATE 26,12:PRINT "NO.:__"
830 LOCATE 30,12:P=VAL(INPUT$(1)
):PRINT USING"#";P
840 LOCATE 31,12:P=P*10+VAL(INPU
T$(1)):PRINT USING"#";P MOD 10
850 CLS:IF P=0 OR P>NP THEN BEEP
:PRINT "PROG. INEXISTENTE":GOSUB
 1270:RETURN
860 GOSUB 970:PRINT "PROG."P
870 LOCATE 2,8:PRINT "TITULO: "P
R$(P)" (";
880 FOR T=1 TO 6:PRINT HEX$(ASC(
MID$(PR$(P),T,1)))SPACE$(1);:NEX
T T:PRINT")"
890 LOCATE 2,10:PRINT "CARGA:"TP
$(TP(P))
900 IF TP(P)-1 GOTO 950
910 LOCATE 2,12:PRINT "DIR:I____
 &H"HEX$(IN(P))"    "IN(P)
920 LOCATE 6:PRINT "F____ &H"HEX
$(FI(P))"    "FI(P)
930 LOCATE 6:PRINT "E____ &H"HEX
$(EJ(P))"    "EJ(P)
940 PRINT:PRINT"  M. O.:"FIX((FI
(P)-IN(P))/1024)"Kb"
950 GOSUB 1270
960 RETURN
970 ' DIBUJA CABECERA
980 CLS
990 PRINT STRING$(39,42)
1000 FOR T=0 TO 2
1010 PRINT CHR$(42)SPACE$(37)CHR
$(42)
1020 NEXT T
1030 PRINT STRING$(39,42)
1040 LOCATE 12,2
1050 RETURN
1060 'DIBUJA CUADRADO
1070 LOCATE 0,8:PRINT SPACE$((38
-X)/2)STRING$(X,45)
1080 FOR T=0 TO Y:PRINT SPACE$((
38-X)/2)CHR$(124)SPACE$(X-2)CHR$
(124):NEXT T
1090 PRINT SPACE$((38-X)/2)STRIN
G$(X,45)
1100 RETURN
1110 GOSUB 1310
1120 LOCATE 0,15:PRINT "NO HALLA
DO":LOCATE 0,15
1130 FOR T=1 TO NP
1140 IF PR$(T)=N$ THEN PRINT "PR
OG. NO.:"T
1150 NEXT T:PRINT
1160 GOSUB 1270
1170 RETURN
1180 LOCATE 17,16:PRINT "(SEGURO
?";
1190 A$=INPUT$(1):IF A$="S" OR A
$="s" THEN NP=0
1200 RETURN
1210 'SELECCION DE OPCION
1220 LOCATE 8,22:PRINT"(SELECCIO
NE OPCION < >)"
1230 LOCATE 28,22
1240 OP=VAL(INPUT$(1)):PRINT USI
NG"#";OP
1250 IF OP=0 OR OP>NO THEN 1230
1260 RETURN
1270 'ESPERA TECLA
1280 LOCATE 12,22:PRINT "PULSE U
NA TECLA"
1290 IF INKEY$="" THEN 1290
1300 RETURN
1310 'ESPERA TITULO
1320 GOSUB 970:PRINT "BUSCA PR."
1330 X=17:Y=2:GOSUB 1060
1340 LOCATE 12,10:PRINT "NOMBRE:
 ______":N$=""
1350 FOR T=1 TO 6:LOCATE 19+T,10
1360 A$=INPUT$(1)
1370 IF ASC(A$)<32 THEN 1360
1380 PRINT A$:N$=N$+A$:NEXT
1390 RETURN
1400 RETURN 150
1410 DATA 13,1A,6F,13,1A,67,E5,C
D,E1,0,CD,E4,0,4F,6,9,C5,CD,E4,0
,C1,B9,28,2,E1
1420 DATA C9,10,F4,FE,D0,28,C,FE
,EA,28,8,FE,D,28,4,FE,D3,20,EC,E
1,77,23,CD,37
1430 DATA DA,E5,CD,E1,0,E1,6,6,E
5,C5,CD,E4,0,C1,E1,77,23,10,F5,C
9
1440
1441 OPEN"cas:arch"FORINPUTAS#1
1442 INPUT#1,NP
1443 FORI=1TO NP
1444 INPUT#1,PR$(I)
1445 INPUT#1,TP$(|P(I))
1446 INPUT#1,IN(I)
1447 INPUT#1,FI(I)
1448 INPUT#1,EJ(I)
1449 NEXT
1450 CLOSE
1460 RETURN
1470 '
1471 OPEN"cas:arch"FOROUTPUTAS#1
1472 PRINT#1,NP
1473 FORI=1TO NP
1474 PRINT#1,PR$(I)
1475 PRINT#1,TP$(TP(I))
1476 PRINT#1,IN(I)
1477 PRINT#1,FI(I)
1478 PRINT#1,EJ(I)
1479 NEXT
1480 CLOSE
1490 RETURN
1499 '          impresora
1500 FOR T=1 TO NP
1510 LPRINT "PROG. NOMBRE  CARGA
 D.INC. D.FIN. D.EJC. MEM.REQ.":
LPRINT
1520 LOCATE0,T+2:LPRINTT
1530 LOCATE7,T+2:LPRINTPR$(T)
1540 LOCATE15,T+2:LPRINTTP$(TP(T
))
1550 LOCATE21,T+2:LPRINTIN(T)
1560 LOCATE28,T+2:LPRINTFI(T)
1570 LOCATE35,T+2:LPRINTEJ(T)
1580 LOCATE42,T+2:LPRINTFIX((FI(
T)-IN(T))/1024)"kb."
1590 NEXT T
1600 RETURN
```

HAUNTED HOUSE

Estamos en una casa embrujada y debemos hallar la salida. Para ello tendremos que sortear toda clase de personajes dignos de una galería del terror: cerebros, esqueletos, fantasmas, pisos que se derrumban, goteras con ácido, etcétera.

Nuevamente, Eurosoft, haciendo gala de un amplio conocimiento de las capacidades gráficas y de sonido de la norma, obtiene unos resultados muy interesantes. Por ejemplo, el estado de nuestro personaje se indica con un esqueleto cuyo corazón (muy bien logrado) late y, a medida que nos hieren, va cambiando el color y la velocidad de los latidos. Si nos matan, la boca de la calavera se abre (presumiblemente dice "¡Ay!").

Aquellos entusiastas de los juegos de aventuras en laberintos, como "Manic Miner " o "Misterio en el Nilo" estarán de parabienes con éste, ya que para resolver cada pantalla se debe demostrar ingenio y velocidad.

En algunos casos un salto bastará para lograr "zafar", pero en otros debemos coordinar nuestro movimiento con el del monstruo de turno (cuyo simple contacto es mortal).

Los controles son sencillos: con los cursores movemos nuestro personaje y vamos recogiendo llaves, fuentes de poder, cruces (para repeler fantasmas), rosarios y demás implementos que nos permitirán luchar contra las fuerzas del mal que controlan la casa.

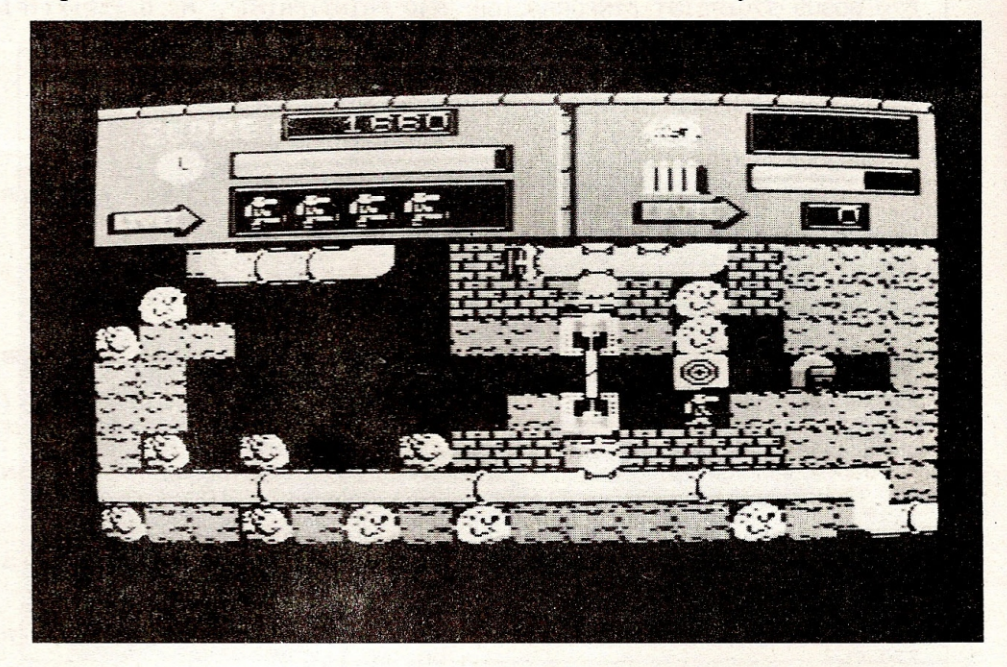
Tiene todos los ingredientes que harán las delicias de los "aventuramaníacos". (Distribuyen: Red Point / Real Time)

BLOW UP

El amigo Joe Kowalski debe cumplir una misión peligrosa: recorrer las galerías subterráneas para solucionar un problema en el tendido del gas de la zona.

Para ello deberá excavar sus propios caminos, enfrentarse a enemigos y seguir avanzando hasta llegar al destino.

Aquellos memoriosos de la norma verán que "Blow Up" es una versión muy sofisticada del conocido "Boulder Dash", con sus galerías, rocas que se deslizan y pueden aplastarnos, algunas zonas con gas "lentificante", que nos hacen avanzar a paso tortuga, etcétera.



Es un juego muy interesante y que realmente atrapa, pues lograr vencer los obstáculos no es nada fácil.

La calidad gráfica y los efectos de sonido son de primer nivel y están acordes con el resto de los juegos pertenecientes a la línea de Eurosoft. (Distribuye: Real Time)

TETRIS

Microsoft pone a nuestro alcance este programa largamente esperado por los amantes de los juegos de ingenio.

A nuestro país llegó primero como un programa de PC y se generó inmediatamente el clan de los "tetrismaníacos".

A simple vista, el juego no promete mucho: piezas geométricas formadas al azar con 4 cuadrados llamados tetróminos (de allí el nombre del programa) caen en cualquier posición. Nosotros debemos ubicar estas piezas en un contenedor a medida que bajan. Si logramos completar una línea de cuadrados, el conjunto desciende una línea y nos libera espacio en el contenedor. Gana quien mantiene más tiempo en funcionamiento el sistema.

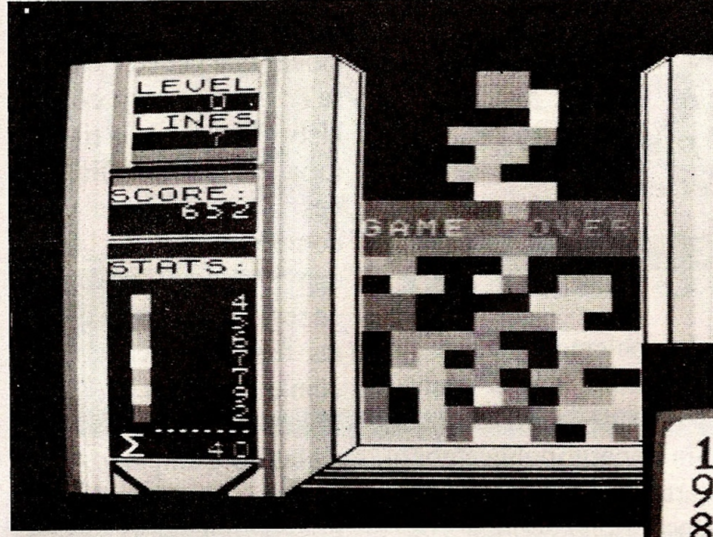
Las piezas se manejan rotándolas y moviéndolas horizontalmente. El puntaje se otorga en función de tres factores: el nivel corriente, la velocidad para acomodar las piezas y si espiamos la pieza siguiente.

Se elige el nivel al comenzar el juego. Luego, el nivel aumenta cada vez que eliminamos 10 filas; pero si, por ejemplo, se comenzó en el nivel 3, se permanecerá en ese nivel hasta que se eliminen 30 filas.

Se puede mejorar el puntaje dejando caer instantáneamente las piezas tan pronto como se las tenga alineadas.

Este juego fue desarrollado en la Unión Soviética por Viktor Gerasimov y Andrei Pajitnov. La primera versión para PC estaba escrita en Turbo Pascal 4.0. Fue traducida al inglés (lástima, hubiera sido lindo verlo escrito en ruso), por David Howorth.

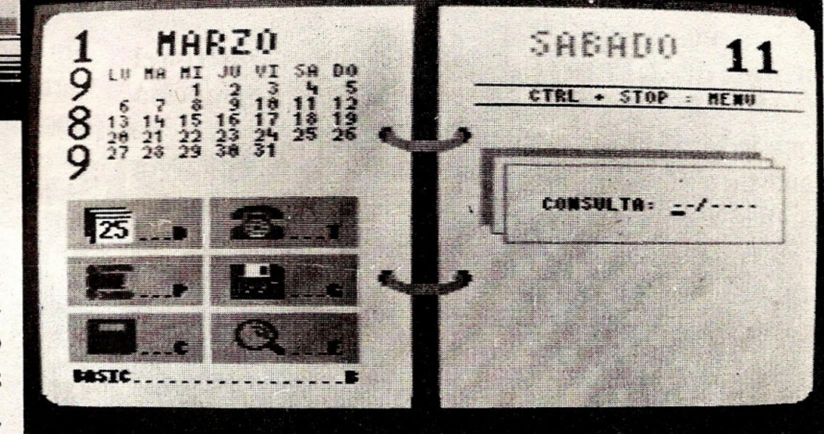
Dicha versión es de dominio público y no puede ser vendida comercialmente. Sólo puede incluirse en BBS's con un archivo de explicaciones.



El archivo de explicaciones rezaba el siguiente mensaje:
Si usted encuentra este juego interesante y lo utiliza, rogamos enviar 20 rublos a:
Víktor Gerasimov
Gorki Av. 212 F 4
Moscow
Soviet Union
¿Alguien sabe a cuántos australes equivale un rublo?
Rublos aparte, la versión de MSX es de primer nivel. Tiene un buen acompañamiento musical, los colores de las piezas están bien definidos y se nos muestra la sumatoria de las piezas colocadas, el estado, el nivel del jugador y cuántas filas completamos.
Los comandos son algo extraños, tanto para juegos MSX en general (no se usan los cursores) y como para la versión PC (las teclas son diferentes a dicha versión) .
Como conclusión decimos que esta versión no defraudará a los ya iniciados. A los principiantes les sugerimos jugar un par de minutos. Si a las tres horas de juego se percatan de que siguen jugando, no digan que no les advertimos... (Distribuye: Real Time)

PORTFOLIO - AGENDA PERSONAL

Basándose en los utilitarios para MSX2 tales como el Philips Home Office o el Philips Ease, surge esta versión de un paquete de agenda personal, telefónica y calculadora para MSX1.



Lo que se nos presenta en pantalla es la imagen del popular "taco" de oficina, en donde generalmente se puede ver el calendario, la fecha de hoy y un espacio en donde se hacen anotaciones y se apuntan las citas. El funcionamiento de este programa es muy similar al taco, con el agregado de la agenda telefónica, calendario perpetuo y calculadora.
Los gráficos del programa son muy ilustrativos, ya que cada opción está simbolizada con un ícono, y se van presentando los distintos mensajes con "ventanas" escritas en 64 columnas (modo gráfico, letra pequeña, como el Tasword).

Las distintas opciones disponibles son:
D - Escribe en la agenda diaria (el texto del taco)
T - Agenda telefónica (muestra agrupado por letra)
P - Seleccionar el mes que se mostrará (con calendario perpetuo)
G - Genera una agenda anual
C - Opción calculadora
E - Permite buscar un texto dentro del taco.
El uso es bastante sencillo, aunque algunas traducciones no están muy de acuerdo con la Real Academia Española (donde dice "Tecle una tecla" debe decir "Pulse . ..") En resumen, un utilitario interesante y práctico para el hogar o la oficina moderna. (Distribuye: Real Time)

LECTOR CON PROBLEMAS

Les quiero hacer conocer una inquietud personal con el fin de que me propongan una solución dentro de sus posibilidades.
Soy lector de MSX desde junio de 1987 y poseo una MSX2, con varios periféricos.
Recientemente compré el software de SUELDOS Y JORNALES producido por ISM (Informática San Martín S.A.) que viene en disquete y con un cartucho llave.
Al aprender el uso del programa comprobé que el mismo liquida e imprime los distintos ítems con milésimas de austral, sin poder cambiarse a centésimas o décimas, es decir, esto no es modificable por el usuario.
Ante este problema me dirigí al comercio donde adquirí el programa. Allí me dijeron que las milésimas no tienen importancia. Yo pienso lo contrario ya que una contabilidad que se registre con centésimas de austral no puede mezclarse una vez al mes con cifras que llevan milésimas. Además, en un futuro no muy lejano probablemente no se van a utilizar más las centenas ni las decenas de austral.
También me dijeron que ésas son las desventaja del "software enlatado" y que ellos no pueden hacer nada, luego me propusieron que "yo" consultara a "su" proveedor.

Para comunicarse con nosotros deben escribirnos a 'Load MSX", Méjico 625 - 3º piso (1097) Capital Federal.

Al consultar al distribuidor me comunicaron que ya no representaban más a ISM y que "perdieron el rastro" de dicha empresa porque se cambió de domicilio. Sin embargo, me prometieron intentar conseguir el nuevo domicilio y telefonearme.
Me pregunto si LOAD MSX puede proponerme algún tipo de solución. No creo que un soft que vale 110 dólares tenga un tiempo de vida tan limitado y sin niguna garantía. El problema que se plantea debió ser previsto por los diseñadores ya que nuestro país se caracteriza por su elevada inflación.
Lamento que mi primera carta a la revista sea por este motivo, pero prometo escribirles nuevamente para dilucidar inquietudes que tengo sobre diversos temas informáticos.

Damián R. Castaño
San Miguel de Tucumán

LOAD MSX

Lamentablemente no te podemos ayudar a resolver el problema, ya que no podemos "meter mano" dentro del soft enlatado. Sin embargo publicamos tu carta para advertir a nuestros lectores sobre los problemas que enfrentás y que no les ocurra lo mismo.

PREGUNTA BOOLEANA

Quisiera que me informen sobre los siguientes términos y conceptos: lógica de Boole, algoritmo, network y CAD. Desde ya, muchas gracias por sacarme de mi ignorancia computacional.

Pablo Uberti
San Fernando

LOAD MSX

La lógica de Boole o binaria es el concepto primario en el diseño de computadoras. Las reglas que rigen el comportamiento de las compuertas de un circuito fueron desarrolladas por el matemático inglés George Boole, a mediados del siglo XIX.
Algoritmo es el método para resolver un problema. Es un conjunto de operaciones que deben ejecutarse en un orden específico.
Network o red es la trayectoria de las comunicaciones entre terminales y computadoras, o entre computadoras. La red puede ser una mezcla de distintas formas de canales de comunicaciones.
Finalmente, CAD son las siglas de Computer Aided Design (Diseño asistido por computadora). Los sistemas CAD se utilizan para diseñar una infinidad de productos industriales y de consumo.

MSX2

Tengo una MSX2 y perdí el manual de la computadora. Quisiera me comunicaran dónde puedo conseguir uno nuevo. De paso una pregunta: ¿cómo puedo acceder a las herramientas que tiene la MSX2?

Martín Cabañas
Tristán Suárez

LOAD MSX

El manual lo podés conseguir en cualquier Centro de Asistencia al usuario.
En cuanto a las herramientas de la MSX2 hay que apretar, simultáneamente las teclas CTRL y SHIFT


COMPUTACION
FEBRERO 1989
AÑO 4 Nº 47 A 4.49 REP. ARGENTINA
K64
PARA TODOS
Con notas de
BYTE
EL FUTURO
PREDICEN
LOS GENIOS
BYTE AWARDS
LO MEJOR PARA PC
PROTECCION
VIRUS, ROBO
Y ACCIDENTE
SOFT INTEGRADO
OPEN ACCESS II
PC IBM COMPATIBLES - DREAN COMMODORE - MSX - SPECTRUM